High-Performance Tsunami Modelling with Modern GPU Technology

Reza Amouzgar

Thesis submitted for the degree of

Doctor of Philosophy

Newcastle University

Faculty of Science, Agriculture and Engineering

School of Civil Engineering and Geosciences

April 2017

Abstract

Earthquake-induced tsunamis commonly propagate in the deep ocean as long waves and develop into sharp-fronted surges moving rapidly coastward, which may be effectively simulated by hydrodynamic models solving the nonlinear shallow water equations (SWEs). Tsunamis can cause substantial economic and human losses, which could be mitigated through early warning systems given efficient and accurate modelling. Most existing tsunami models require long simulation times for real-world applications. This thesis presents a graphics processing unit (GPU) accelerated finite volume hydrodynamic model using the compute unified device architecture (CUDA) for computationally efficient tsunami simulations. Compared with a standard PC, the model is able to reduce run-time by a factor of > 40.

The validated model is used to reproduce the 2011 Japan tsunami. Two source models were tested, one based on tsunami waveform inversion and another using deep-ocean tsunameters. Vertical sea surface displacement is computed by the Okada model, assuming instantaneous sea-floor deformation. Both source models can reproduce the wave propagation at offshore and nearshore gauges, but the tsunameter-based model better simulates the first wave amplitude.

Effects of grid resolutions between 450-3600 m, slope limiters, and numerical accuracy are also investigated for the simulation of the 2011 Japan tsunami. Grid resolutions of 1-2 km perform well with a proper limiter; the Sweby limiter is optimal for coarser resolutions, recovers wave peaks better than minmod, and is more numerically stable than Superbee. One hour of tsunami propagation can be predicted in <1 minute using 1350 m or coarser resolutions. Run-time is reduced by >50 times on a regular low-cost PC-hosted GPU, compared to a single CPU. For 450 m resolution on a larger-memory server-hosted GPU, performance increased by ~70 times.

Finally, two adaptive mesh refinement (AMR) techniques including simplified dynamic adaptive grids on CPU and a static adaptive grid on GPU are introduced to provide multi-scale simulations. Both can reduce run-time by ~3 times while maintaining acceptable accuracy. The proposed computationally-efficient tsunami model is expected to provide a new practical tool for tsunami modelling for different purposes, including real-time warning, evacuation planning, risk management and city planning.

Acknowledgments

I would like to express my deepest gratitude to my supervisors, Prof. Qiuhua Liang and Prof. Peter Clarke, for their patience, invaluable advice, constructive comments, guidance, and encouragements. My supervisors have enabled me to fulfil this thesis and it has been my honour to be their student. When I was searching for a PhD project I was very delighted when I realised Prof. Liang's research is about computational hydraulics at Newcastle University and part of his active research and interest is related to natural hazards which was exactly what I was looking for. Among the available research program I have been always keen to study tsunami modelling and contribute in this field of knowledge so if I can be of any help for those people around the world who might be vulnerable to this disastrous event.

I would also like to thank Dr Stephen McGough (Durham University) and Dr William Blewitt (Newcastle University) from school of computing sciences for their seminar and talks about high-performance computing.

After model development I had the opportunity to visit the disaster prevention institute (DPRI), Kyoto University of Japan to undertake the preliminary 2011 Tohoku tsunami modelling. In this regard I would like to give special thanks to Dr Tomohiro Yasuda and Prof. Hajime Mase for their valuable assistance providing necessary data and information during my academic visit.

I would also like to extend my thanks to peers, colleagues, and researchers who I met during my work so I could ask my questions, discuss and share our views including Hongbin Zhang, Xilin Xia, Jingming Hou, Jingchun Wang, Luke Smith, and Ciprian Spatar. My thanks also go to the administrative staff in School of Civil Engineering and Geosciences, SAgE Faculty workshop organisers, librarians and computing technicians for the services they provided especially Melissa Ware, Hannah Lynn, Graham Patterson and Iain Woodfine.

Finally I would like to express my sincere gratitude to my parents for their continued love and support, I owe my achievements to them. Without their continuous help and support I could not be able to start and continue for my post graduate studies in overseas.

Abbreviations

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
AMR	Adaptive Mesh Refinement
API	Application Processing Interface
CFL	Courant-Friedrichs-Lewy
COMCOT	Cornell Multi-grid Coupled Tsunami Model
COULWAVE	Cornell University Long and Intermediate Wave Modelling Package
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DART	Deep-ocean Assessment and Reporting of Tsunamis
DCRC	Disaster Control Research Centre
DG	Discontinuous Galerkin
DP	Double Precision
DPRI	Disaster Prevention Research Institute, Kyoto University, Japan
DRAM	Dynamic Random Access Memory
FDM	Finite Difference Method
FEM	Finite Element Method
FUNWAVE	Fully Nonlinear Boussinesq model
FVM	Finite Volume Method
GEONET	GPS Earth Observation Network
GITEWS	German-Indonesian Tsunami Early Warning System

GloBouss	Global Boussinesq Solver
GPS	Global Positioning System
GPU	Graphics Processing Units
HLL	Harten, Lax, van Leer
HLLC	Harten, Lax, van Leer-Contact
HPC	High-Performance Computing
JMA	Japan Meteorological Agency
LES	Large Eddy Simulation
LSWE	Linear Shallow Water Equations
MOST	Method of Splitting Tsunami
MPI	Message Passing Interface
MUSCL	Monotonic Upstream-Centered Scheme for Conservation Laws
NLB	Non-Linear Boussinesq model
NLSWE	Non-linear Shallow Water Equations
NOAA	National Oceanic and Atmospheric Administration
NVIDIA	An American global Technology Company
OBP	Ocean Bottom Pressure Gauges
ODE	Ordinary Differential Equation
OpenCL	Open Computing Language
OpenMP	Open Multiple Processing
PC	Personal Computer
POM	Princeton Ocean Model
RAM	Random Access Memory
RANS	Reynolds Averaged Navier-Stokes

RMSE	Root-Mean-Square Error
SFC	Sierpinski space filling curves
SMF	Submarine Mass Failure
SP	Single Precision
SPH	Smooth Particle Hydrodynamics
STL	Standard Template Library
SWE	Shallow Water Equations
TIME	Tsunami Inundation Modeling Exchange
TsunAWI	Alfred Wegener Institute for polar and marine research
TUNAMI	Tohoku University's Numerical Analysis Model for Investigation of tsunamis
TVD	Total Variation Diminishing
UCSB	University of California, Santa Barbara
UN	United Nations
USGS	United States Geological Survey
VOF	Volume of Fluid

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Introduction and motivation of the research

Tsunamis are gravity waves, wave trains or series of waves which are produced by a sudden large displacement of a body of water. Tsunamis could be caused by any disturbances below or above the water surface that displace a bulk amount of water from its equilibrium. The causes of tsunamis include submarine earthquakes, landslides, collapses of volcanic edifices, eruptions of submarine volcanoes or even meteorite impacts. Tsunamis can be very destructive and may cause huge loss of lives. One way to mitigate potential loss of lives from similar events would be through operation of effective early warning systems and quick evacuation of vulnerable population in the affected coastal areas to safer places, which requires a computationally efficient and reliable tsunami model to predict in real time tsunami wave propagation and inundation. In addition to this, tsunami numerical modelling provides an indispensable tool for risk assessment and management.

1.1.1 *Tsunami effects and tsunami modelling*

Tsunamis caused by earthquakes are the most destructive among tsunamis from different sources, and are the focus of this research. Tsunamis are a type of natural hazards of medium probability and potentially high risk to coastlines, with catastrophic effects (Bernard *et al.*, 2010). Mega-tsunamis used to be considered as an extremely rare phenomenon and they have attracted less attention by communities and scientists. However, after the 2004 Indian Ocean tsunami, this type of natural hazards has become highlighted to the world, both public and research communities. This tsunami hit 14 countries severely and caused more than 230,000 deaths, subsequently known as the deadliest tsunami in history. Just a few years later another mega-tsunami was initiated by the 2011 Tohoku-Oki earthquake, which killed over 16,000 people with an extra 2,500 reported missing. Table 1.1 presents the recent major tsunamis that occurred between 2004 and 2011 with approximate numbers of casualties.

According to a new report by United Nations (UN), each year about 60,000 people and four billion US dollars in assets are exposed to global tsunami hazard (Desai *et al.*, 2015). Also, tsunamis can cause other indirect negative effects such as disease, destructions, environmental impacts, and psychological effects. Owing to the growth of population, immigration to coastal regions, and climate change, fatalities and damages caused by extreme natural hazards including tsunamis are estimated to increase over time (Huppert and Sparks, 2006).

It is therefore crucial to better understand tsunami risk and implement effective risk management strategies for risk reduction. Accurate and reliable tsunami warning systems have been demonstrated to play an important role in mitigating tsunami impacts and protecting people living in the affected areas (Bernard and Titov, 2015).

Recent tsunami events	Number of losses
2004 Indian Ocean	230,000
2006 Java	800
2009 Samoa islands	180
2010 Chile	500
2010 Sumatra	400
2011 Japan	16,000
2015 Chile	13

Table 1.1: Major tsunami events occurred in the last 1 ~ 2 decades and the approximate number of deaths.

Since the 2004 Indian Ocean tsunami there has been continuous research in tsunami and has led to rapid development in the context of computer modelling for the entire life span of a tsunami, including generation, propagation and inundation. However, more research is still needed to further improve the current tsunami models, in terms of both computational efficiency and accuracy. This is evident following the 2011 Japan tsunami, for which the current models failed to operate as successfully as was expected. For example, in the previous tsunami-hazard assessments, the large tsunami at the Fukushima nuclear power station, which caused the world's worst nuclear accident (Satake *et al.*, 2013), was not accounted for accurately.

Tsunami mathematical models and numerical schemes to solve those models can be useful in different ways (George, 2006). First, they can be utilized for potential flood mapping to inform emergency planning and design and construction of coastal infrastructure, which are essential for reducing the hazards. Second, research communities can use these tools to study the physical process of tsunami generation, propagation and inundation. These studies can provide better understanding and enable better forecasting of potential tsunami events. Moreover, numerical modelling can be applied to create a seismic benchmark for issuing tsunami warnings if a real tsunami is triggered. Finally, real time simulations can be used in tsunami warning systems. Therefore, numerical simulations are beneficial for studying the tsunami impacts of past events, for evaluating the tsunami hazard, or creating a pre-computed tsunami database. However, it can be very challenging to implement the numerical tools for tsunami forecasting in real time due to the expensive computational time (Power, 2013).

Tsunami models can be categorized as physically based numerical models and empirical models. Numerical models solve the fluid dynamics governing equations of the tsunami through a grid system where the geographic features such as bathymetry and topography are involved in the simulations and the tsunami wave height, arrival time, speed and inundation depth can be predicted.

In empirical models, tsunami properties can be evaluated by employing statistical relationships from field and/or simulation data (Power, 2013). These models are fast to compute but they are limited in estimating the tsunami characteristics because they do not consider physical properties and mainly rely on a simple equation achieved by statistical data-fitting approaches. In contrast to numerical models, empirical models are incapable of explicitly representing the interaction between the tsunami waves and complex bathymetry/topography and predicting arrival time. However, empirical models can provide a rapid evaluation of the tsunami height especially when there is insufficient information available for the source or the high-resolution bathymetry/topography are not available. For example Abe (1979) proposed an algebraic equation based on compilation of historic data to estimate the tsunami height at a distant coast by incorporating the earthquake magnitude.

1.1.2 *Brief history of numerical tsunami modelling*

Tsunami research started to develop in Japan because Japan has been affected by earthquakes and local or distant tsunamis much more frequently compared to most other countries. The 1896 Meiji-Sanriku earthquake, one of the most destructive earthquakes in Japan's history, caused two tsunamis and over 22,000 deaths. In 1933, 37 years later, another earthquake at approximately the same location generated a tsunami which hit Sanriku region. This tsunami destroyed over 7,000 houses, and the death and missing toll were reported to be more than 3,000 people. After the 1933 Sanriku tsunami, disaster mitigation including hard and soft countermeasures was proposed with the 'modern' knowledge at that time, for example relocation of houses to a higher level, and constructing coast dykes and sea walls. The first tsunami forecasting started in 1941, and details can be found in (Shuto and Fujima, 2009).

Mansinha and Smylie (1971) developed a model to predict the deformation of the seafloor caused by an earthquake which could therefore initiate a tsunami. This was the first step for numerical simulation of the tsunami. However, to simulate a real tsunami event the model needed computer resources and technologies at a large scale which was not available at that time. Hence, since the late 1970s the computer modelling of tsunamis started developing along with computer technology advancements. In this regard Goto and Ogawa (1982) proposed a

numerical method to simulate the tsunami process from generation to run-up on land. Based on linear wave theory, they used linear shallow water equations in the deep sea and considered non-linear shallow water equations for the shallower coastal region and inundation. They implemented a finite difference upwind scheme on a staggered grid in space and a leap-frog explicit scheme in time. Model development continued by Japanese tsunami modelers, e.g. (Shuto *et al.*, 1990; Imamura, 1996; Goto *et al.*, 1997). Then in 1999, this model became the basis of Japan Meteorological Agency (JMA) for numerical forecasting. It is also reported that the aforementioned tsunami model was one of the first implementations of rapid numerical modeling to tsunami forecasting in the world; details can be found in (Tatehata, 1997). The model is now well-known worldwide as the TUNAMI package and is authored by Professor Imamura in the Disaster Control Research Centre in Tohoku University through the Tsunami Inundation Modelling Exchange (TIME) program which is distributed to 43 institutes in 22 countries.

Since the past three decades computing technologies and numerical techniques have been developing in parallel rapidly. There are now many numerical packages available for tsunami simulations and some of which will be reviewed in the next chapter. There are similarities and differences between these models in terms of the mathematical assumptions and the numerical schemes adopted to solve the governing equations. Therefore, the capabilities of these models, and their computational efficiency might be different. Among different types of tsunami models, the 2D shallow water equations (SWE) and Boussinesq models are most commonly used for tsunami simulations. However, SWE models are more popular for full life-span tsunami simulations because they need less computational effort compared to other models, whereas the simulated results are relatively reliable.

In order to solve the mathematical formulation (e.g. the SWE), a conventional finite difference scheme has been mainly used in the past (Shuto *et al.*, 1990; Liu *et al.*, 1995a; Titov and Synolakis, 1995). The finite difference approaches are numerically easier to implement and computationally more efficient than other numerical methods, e.g., finite element methods. However, finite element models using unstructured grids can deal with complex geometry at their boundaries but this advantage is less important for tsunami modeling in the open ocean. Another numerical method that has gained increasing popularity in the last two decades is the finite volume method that solves the integrated form of the governing equations. Finite volume models are fully conservative and can be implemented on unstructured meshes for boundary fitting. When incorporated with a Godunov-type scheme, finite volume models can handle different types of flow regimes including subcritical, transcritical and supercritical and

automatically capture flow discontinuities, such as shocks and bores. Therefore the tsunami wave evolution from propagation in the deep ocean to inundation on land, which will face different flow/wave features such as shoaling, wave breaking, and bores, can be reliably estimated by a finite volume Godunov-type model that solves the SWE or Bussinesq equations. Also, the computational efficiency of finite volume schemes is generally comparable to the finite different methods. Examples of this type of tsunami models include those reported by Berger *et al.* (2011) (GeoClaw software) and Popinet (2011).

1.2 Tsunami source model

As mentioned previously, tsunamis can be generated by various types of sources below or above the sea surface, although this work focuses only on the tsunamis initiated by earthquakes. The initial conditions to set up a tsunami hydrodynamic model are obtained from the estimated coseismic sea floor deformation. A reliable and efficient tsunami simulation is therefore highly dependent on the accurate and fast estimation of the source model. Recognizing that most uncertainties for tsunami modelling arise from the source model, the research for tsunamigenic earthquakes has been under continuous development and improvement. The tsunami source can be determined by different methods based on teleseismic data, tsunami wave, tsunameters (DART buoy), and more recently using GPS technology. More details about the source models are reviewed in next chapter and their implementation will be demonstrated in Chapter 5.

It should be mentioned that the research relevant to the earthquake source and early detection of the tsunami wave is very important, but it is related more to the different fields of research such as geophysics and geodesy. This work only implements the source as an application to initialize the hydrodynamic model. In this regard, two different source models as provided in the literature are employed to reproduce the 2011 Japan tsunami (Fujii *et al.*, 2011; Wei *et al.*, 2013), which will be discussed in details in the later chapters.

1.3 High-performance computing

Although tsunami models based on the SWEs are computationally more efficient compared with those based on Boussinesq or 3D fluid dynamics equations, they are still computationally too demanding for real-time tsunami forecasting. High-performance computing (HPC) techniques have been developed to advance computational fluid dynamics and also applied to speed up tsunami models.

Through better harnessing computer power, parallelization has been used to achieve high-performance computing for decades. Parallel computing may initially refer to distributed and

shared memory parallelization which are mainly employed for computer clusters or supercomputers. Most desktop and laptop computers are now equipped with multi-core processors facilitating shared memory parallelization. Multi-core processors can potentially scale up the speed of the simulation depending on the available number of cores. For example Son *et al.* (2011) parallelized the hybrid COMCOT tsunami model in combination with a Boussinesq model through implementation of OpenMP for a shared memory system. The resulting parallelized code shows good linear scale-up performance and the computational time was reduced by nearly eight times by employing eight processors.

Parallelization using distributed memory architecture is also common as different processors are used as part of a large cluster. The computational domain is divided into sub-domains and each domain is assigned to a separate processor for computation. This is useful for very large-scale simulations to overcome the memory limitations. This technique requires data communication between processors through a message passing interface (MPI) which is relatively difficult to implement and there are some problems which may arise from using this technique such as load balancing (Grudenić *et al.*, 2008; Vijayasherly *et al.*, 2016).

More recently, modern graphics processing units (GPUs), which were first developed for rendering real-time effects in computer games, have been explored to provide significant computational power for scientific computing. Since 2010 research on GPU implementation for scientific computing is increasing, and this is now an area of active research beside other parallel acceleration techniques. Algorithms that present the same calculation with different data can benefit from single instruction multiple thread (SIMT) processors available on GPUs to deliver performance which is often tens of times faster than the same calculation on a multi-core CPU (Brodtkorb *et al.*, 2012). The GPU computing revolution can be interpreted as an evolution from traditional multi-core CPUs to many-core graphic processors. It is expected in future more scientific packages will be developed or updated to be compatible for running on GPUs.

Attempts have been made in the last few years to implement SWE models on GPUs for different applications (Smith and Liang, 2013; Vacondio *et al.*, 2014; Nishiura *et al.*, 2015; Juez *et al.*, 2016). In the context of tsunami modelling, the tsunami wave propagation of the MOST tsunami model is accelerated by different programming platforms including OpenMP, CELL architecture and GPUs (Vazhenin *et al.*, 2013). However, the development of GPU-accelerated high-performance tsunami models is still at the beginning and there is substantial scope for further improvement, which is the focus of this thesis.

1.4 Aim and objectives

The aim of this work is to develop a novel, computationally efficient tsunami model by harnessing the latest GPU technology. This aim will be achieved through the following objectives:

- A finite volume, Godunov type SWE model will be improved to allow parallel computing through OpenMP and validated for tsunami modelling including wave propagation and inundation.
- The finite volume, Godunov-type SWE model will be implemented for fully parallelized computing on GPUs using the Compute Unified Device Architecture (CUDA) programming framework.
- Numerical experiments will be undertaken to evaluate the performance of the improved parallelized models on different hardware devices.
- Apply the high-performance tsunami models to reproduce the 2011 Japan tsunami and test sensitivities of the numerical solutions to model parameters.
- Further develop the GPU-accelerated tsunami model to using adaptive mesh refinement to improve further its computational efficiency.

1.5 Thesis outline

Chapter 1 briefly introduces the background of the tsunami research, and discusses the importance of tsunami modelling. The role of high performance computing in tsunami modelling is highlighted as the aim of this project, followed by the objectives to achieve this aim.

Chapter 2 presents a comprehensive literature review in the context of tsunami simulation, including hydrodynamic tsunami models, numerical schemes, acceleration techniques for hydrodynamic models and the tsunami earthquake source models.

Chapter 3 introduces the methodology employed in this work, i.e. the adopted hydrodynamic model and the GPU implementation.

Chapter 4 validates the GPU accelerated model against classical tsunami benchmarks. The speed up of the parallel model is evaluated through comparison with the sequential version running on a CPU.

Chapter 5 focuses on the case study of the 2011 Japan tsunami.

Chapter 6 introduces the simplified adaptive mesh refinement techniques used on CPU and GPU for tsunami simulations.

Chapter 7 draws conclusions out of the current work and proposes future work in continuation of this research.

Publications up to date

Parts of the work presented in this thesis have been published and presented in three journal papers and four conference papers.

Journal papers

Amouzgar R, Liang Q, Smith L.A (2014) *GPU-accelerated shallow flow model for tsunami simulations*, Proceedings of the ICE - Engineering and Computational Mechanics 167, 117-125.

Amouzgar R, Liang Q, Clarke PJ, Yasuda T, Mase H (2016) '*Computationally Efficient Tsunami Modelling on Graphics Processing Units (GPU)'*, International Society of Offshore and Polar Engineers, 26(2), pp. 154–160.

Liang Q, Hou J, **Amouzgar R** (2015) 'The simulation of tsunami propagation using adaptive Cartesian grid, Coastal Engineering Journal', 57 (04).

Conference articles

Amouzgar R, Liang Q (2014) '*Application of a GPU Based Hydrodynamic Model in Tsunami Simulations'*, the Twenty-fourth International Ocean and Polar Engineering Conference, 15-20 June, Busan, Korea.

Amouzgar R, Liang Q (2015) '*Tsunami Simulations on Different Hardware Devices'*, workshop on advances in numerical modelling of hydrodynamics, Sheffield University, March 2015,

Liang Q, **Amouzgar R** (2016) '*Performance of Difference Hardware Devices for Tsunami Simulations'*, the 26th International Ocean and Polar Engineering Conference, June, 15-20 June 2016, Rodos, Greece.

Xiong Y, Liang Q, **Amouzgar R**, Cox DT, Mori N, Wang G, Zheng J (2016) '*High- Performance Simulation of Tsunami Inundation and Impact on Building Structures'*, the 26th International Ocean and Polar Engineering Conference, June, 15-20 June 2016, Rodos, Greece.

Chapter 2 Literature Review

This chapter first reviews different types of hydrodynamic models in the context of tsunami modelling, followed by different numerical schemes used in these hydrodynamic models. The advantages and limitations for each model in terms of accuracy and computational efficiency is highlighted where possible, aiming to select a suitable model and numerical scheme for the current work. The tsunami hydrodynamic models, even those solving the two-dimensional shallow water equations, suffer from long computational time, limiting their application for real-time tsunami simulations. Therefore different acceleration techniques, including high-performance computing in the context of hydrodynamic modeling, are reviewed. In this regard a new GPU based hydrodynamic model is later proposed in this work (Chapter 3). Then the tsunami source models generated by earthquakes are also discussed briefly to explain the capability and limitation of different methods. As a key objective of this thesis is to validate the proposed model by reproducing the 2011 Japan tsunami, some of the recent researches focusing on this mega-tsunami are also reviewed.

2.1 Tsunami models and numerical schemes

Tsunamis have been historically and commonly modeled by the shallow water equations (SWE) (Shuto *et al.*, 1990; Liu *et al.*, 1995a; Titov and Synolakis, 1995; Goto *et al.*, 1997). It should be noted that based on linear wave theory, shallow water waves are applicable for the cases where the horizontal length scale (wave length) is much larger than the vertical length scale (water depth), (h<0.05 λ), which is shown schematically in Figure 2.1.

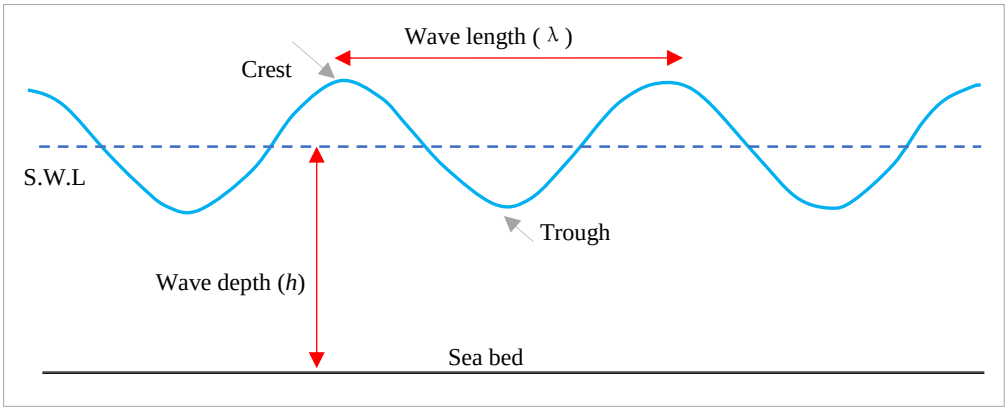


Figure 2.1 Schematic view of linear wave characteristics.

However, Boussinesq-type models, which represent an extension to the shallow water equations, are also popular and widely used as they can better describe the wave dispersions (Kirby *et al.*, 1998; Lynett and Liu, 2004; Løvholt *et al.*, 2008). Different numerical schemes have been developed and implemented to solve the governing equations to facilitate tsunami modeling,

which include finite difference method (FDM), finite volume method (FVM), finite element method (FEM) and smooth particle hydrodynamics (SPH). Some of these tsunami models and the corresponding numerical schemes are reviewed in this section.

Kirby *et al.* (1998) and Wei *et al.* (1995) developed a fully nonlinear Boussinesq wave model known as FUNWAVE. Chen *et al.* (2000) used FUNWAVE to simulate wave transformation and breaking, with an objective to implement and validate an extended time-domain Boussinesq model for wave transformation resulting from combined refraction and diffraction, wave breaking, and wave runup in two horizontal dimensions. Ioualalen *et al.* (2007) and Grilli *et al.* (2007) studied the 2004 Indian ocean tsunami with FUNWAVE and a non-linear shallow flow model at a coarse resolution (1′, ~1800 m) and a finer resolution (0.25′, ~450 m). Dispersive effects were quantified and it was concluded there was a slight difference in the surface elevation between the two models. Depending on the grid size, the depth of the ocean and the distance which the wave propagates, the maximum dispersion effect was found to be up to 20 %. It was argued that the use of a fully nonlinear Boussinesq equation model is superfluous in the context of large-scale tsunami model and it is computationally more expensive.

Kirby *et al.* (2009) reported the effects of dispersion and Coriolis forces by an idealized case on a sphere, which could be dependent on the width of the tsunami source. For example, for a 100 km fault, the Coriolis effect is as important as the dispersive effect, and it gets relatively more important for a source with a larger width. The sensitivity to dispersion and Coriolis effects was studied further in simulations for the Tohoku 2011 tsunami (Kirby *et al.*, 2013). It was shown that, for the 2011 Japan tsunami, the dispersion has more effect on the distribution of the wave height at far field. In general, dispersive models predict lower wave height in the near field. In contrast, this effect is reversed in the far field where the estimated wave heights are larger in the dispersive case compared to the standard non-dispersive case. The Coriolis effect was shown to be insignificant and only affected up to 5 % of the wave height, consistent with the simulation reported by Løvholt *et al.* (2008) for the Cumbre Vieja volcano event.

In another study, Horrillo *et al.* (2006) carried out a numerical simulation for the 2004 Indian ocean tsunami by different numerical models including the nonlinear shallow water (NLSWE) (nondispersive), nonlinear Boussinesq (NLB), and the 3D full Navier-Stokes by the volume of fluid (VOF) method. First the model was applied to reproduce a simplified case for a 1D channel to visualize the dispersion effects in a 3D model compared to NLSWE and NLB approaches. It was shown that both the NLB and 3D models could reproduce detailed features such as the distributions of the leading dispersive wave better than the NLSWE. The other important

features of the wave such as amplitude and phase reproduced by the NLSWE were comparable to the two more complicated models. The 2D dispersion effect was also studied in the Bay of Bengal basin through comparison at different gauges. Again a very good agreement was achieved between the non-dispersive and dispersive models. The numerical results further showed that, for gauges near to the tsunami source, the agreement was excellent as the dispersive waves had a short time to develop while the farther gauges were affected more by dispersion and the NLSWE overestimated the lead wave amplitude by about 20%. The study was extended to study the effect of dispersion on run-up estimation at different locations from near to far field compared to the source. The numerical experiments showed that both NLSW and NLB could predict the maximum run-up close to the observations. The run-up height at most coastal region predicted by the two models matched very well. However, after about three hours' propagation in the coast of Sri Lanka, the NLB model reproduced a higher run-up, up to 60% compared to the NLSWE counterpart. The CPU computational time required by the three different numerical schemes was reported for a one-dimensional case. The simulation took the NLSWE model 30 minutes, NLB model 5 hours and the fully 3D model 72 hours to complete. In conclusion, the NSWE model could reproduce the same tsunami event with a good level of accuracy compared to the other two models but at a very low computational cost.

There is another type of shallow water equation which considers dispersion by assumption of non-hydrostatic pressure in the 2D depth average model e.g. (Stelling and Zijlema, 2003; Yamazaki *et al.*, 2009). In this type of model the hydrostatic and non-hydrostatic pressures are decomposed, and the non-hydrostatic component can be solved by an implicit scheme through the three-dimensional continuity equation. The momentum conserved advection scheme with an upwind flux approximation is used to deal with discontinuities and bores. Yamazaki *et al.* (2009) stated, the use of the upwind flux approximation of the surface elevation in the continuity and momentum equations in their model can further improve the model stability which is essential for wave breaking waves. They also reported that their proposed non-hydrostatic model, which was solved by a semi-implicit finite difference scheme, is equivalent to a classical Boussinesq model for weakly dispersive waves. These non-hydrostatic shallow water models may be computationally competitive compared to extended Boussinesq-type wave models, though they cannot be computationally as efficient as classical SWE models for large computational problems in practical applications. The model proposed by Yamazaki *et al.* (2009) was then extended for multi-scale tsunami simulations with a two-way grid nesting technique, known as NEOWAVE (Non-hydrostatic Evolution of Ocean WAVE) (Yamazaki *et al.*, 2011a). Tsunami generation by dynamic seafloor deformation is a possibility of the

NEOWAVE model due to the inclusion of non-hydrostatic pressure and vertical velocity in the governing equations. The application of the NEOWAVE model for reproducing the 2011 Japan tsunami will be reviewed later in this chapter.

Kim and Lynett (2010) used a model based on the fully nonlinear Boussinesq-type equations with a weakly non-hydrostatic pressure assumption to study the dispersive and non-hydrostatic pressure effects at the front of a surge. The equations are solved by a fourth-order accurate finite volume method incorporated with an approximate Riemann solver for the leading order of shallow water terms. For higher order terms the cell averaged finite volume method is implemented. Then, the third order Adams-Bashforth predictor and the fourth order Adams Moulton corrector approach are used. The laboratory tsunami benchmark of Monai was chosen to show the model capability. The simulated results showed that the shape of secondary wave pattern with dispersive properties were the only observed difference compared to the results of the same benchmark problem using SWE models.

Kazolea *et al.* (2012) presented an unstructured finite volume numerical scheme for extended 2D Boussinesq-type equations of Nwogu (1993) in conservation law form. The authors stated that this is the first time the finite volume approach is applied in extended Boussinesq-type equation for unstructured triangular meshes. These meshes have the advantage for dealing with complex geometries and for adapting to the physical scale of the simulations which potentially can lead to significant reduction in the computational cost. The model is validated for several benchmarks and the computational time is reported for the conical island test case (Liu *et al.*, 1995a). It was shown that the computational run time for the 20 s simulation with 52, 191 nodes was approximately 33 min on a single Q6600 CPU. Although the capability of the dispersive model on unstructured grids is presented, but the computational time for a small scale test case shows that the model requires a significant computational cost which might not be appropriate for real-time tsunami simulations.

Glimsdal *et al.* (2013) studied the importance of dispersion for tsunamis generated by earthquakes and landslides. The authors found that moderate-magnitudes (~8 or less) can generate more dispersive tsunamis compared to the severe ones such as the 2004 Indian Ocean and the 2011 Japan tsunami. For large earthquakes the frequency dispersion may slightly modify the transoceanic propagation. Therefore, dispersion is not necessary for near-field propagation. However, if far-field data are used for verification of source characteristics, dispersion might become an important factor. On the other hand, the tsunamis which are induced by landslides are severely affected by dispersive effects. These effects are important

during the wave generation and wave propagation for the leading part of the wave. The authors concluded that for near-field hazard assessment the SWE is a better option for real-time tsunami predictions and for warning purposes. However, during the shoaling process, the undular bores may evolve which are not included in the SWE. Although, these bores may amplify the wave height, but their effects on inundation is not clear as the wave breaking process may strongly dissipate the individual short crests. The conclusion of their work can support the option of using SWE and neglecting the dispersion effects of this research for the near-field simulation of the 2011 Japan tsunami.

Lannes and Marche (2015) introduced a new class of fully nonlinear and weakly dispersive Green-Naghdi models (or Serre, or fully nonlinear Boussinesq) equations. This proposed model share the same order of precision as the standard model but take advantage of a mathematical structure which makes it more suitable for the numerical resolution and computationally it is more efficient. The new developed model relies on hybrid finite volume and finite difference splitting approach. Finite volume scheme solves the hyperbolic part, while the dispersive part is handled with finite difference approach. The robustness of the model is shown through simulation of several tsunami benchmarks for wave propagation, shoaling, diffraction, refraction, and run-up. In the aforementioned study, the computational efficiency of the proposed model compared to the SWE for the conical island test case (Liu *et al.*, 1995a) is also reported. It is mentioned that the 30 s of wave propagation took 510 s with their new proposed model on a single Intel i5 CPU, while the SWE model simulation time was 360 s under the same simulation condition. The comparison of the computational time showed that the new dispersive model approximately consumed 30 % of the whole computational time for the dispersive step. Although, the robustness and efficiency of the model has been presented by the authors for some tsunami benchmarks, but further investigations for real world tsunami simulations are necessary for practical applications.

Popinet (2015) presented a novel combination of methods for the numerical approximation of solutions to the Serre-Green-Naghdi (SGN) equations. In this work, the shallow flow finite volume solver of Popinet (2011) is combined with a generic multigrid solver defining the SGN momentum source term which is suitable to be implemented on adaptive quadtree grids. It was shown that the dispersive effect can become important for deep-ocean propagation of tsunami waves consistent to the results reported by Kirby *et al.* (2013). The study of fine-scale flooding of 2011 Tohoku tsunami, showed that inclusion of SGN dispersive terms did not produce major, systematic changes for the Japan coast. However, it was argued that the inundation of the land for the far field coast (e.g. North America in this case) can be influenced by the details of deep-

ocean propagation and dispersions. In terms of computational efficiency, it was demonstrated that the SGN dispersive model was approximately 4-6 times more expensive than the SWE solver. The physical time of 400 minutes for the 2011 Tohoku tsunami on 8 Intel i7 cores by the aforementioned adaptive quadtree SGN model has been also monitored. In this regard, the run time concerning 280,000 points was reported to take about 160 minutes. The authors stated that the computational performance of the CPU shared memory parallelism is limited, and they are working on distributed-memory MPI parallelism to develop an efficient dispersive tsunami model coupled with adaptive grids on large parallel systems.

TUNAMI, MOST, and COMCOT are popular packages in the tsunami modelling field which have been validated successfully for various benchmarks and historical tsunami events. These models are provided for research groups around the world for tsunami simulation. The TUNAMI code (Tohoku University's Numerical Analysis Model for Investigation of near-field tsunami), which consists of different versions, solves the linear or nonlinear SWEs by a finite difference leap-frog scheme (Shuto *et al.*, 1990; Imamura, 1996; Goto *et al.*, 1997). The model is cross tested and verified with recent work reported for the 2011 Tohoku-Oki tsunami (Oishi *et al.*, 2015).The other widely used model that solves the same equations with relatively similar discretization and nesting methods, is the MOST (Method Of Splitting Tsunami) model of NOAA, developed at University of Southern California (Titov and Synolakis, 1995; Titov and Synolakis, 1998; Synolakis *et al.*, 2008). The model is capable of simulating three processes of tsunami including earthquake, transoceanic propagation and inundation. This model also has been broadly tested against different laboratory experiments and could successfully reproduce many historical tsunamis and was recently used to study the 2011 Japan Tsunami (Wei *et al.*, 2014). The numerical model COMCOT (COrnell Multi-grid Coupled Tsunami model) is based on a modified leapfrog scheme which solves the non-linear and linear shallow water equations (Liu *et al.*, 1995a; Liu *et al.*, 1998) and the model has been used to investigate historical tsunami events e.g. the 1960 Chilean tsunami and 2004 Indian ocean tsunami (Liu *et al.*, 1995b; Wang and Liu, 2006).

Liu *et al.* (2009) presented a comparison of the linear and nonlinear shallow water equations for the 2006 China Sea tsunami. Linear shallow water equations are more simplified, neglecting the friction term, and the nonlinear convective terms. These equations are the simplest form of tsunami equations for tsunami wave propagation, and require less computational effort compared to the nonlinear set of shallow water equations. Liu *et al.* (2009) reported that non-linear shallow water model was about 4.5 times computationally more expensive compared to the linear shallow water model. The comparison of linear and nonlinear predictions of tsunami

wave propagation across the China Sea reveals that the linear shallow water equation is valid for the South China Sea simulation but this is not the case for the eastern China Sea due to its much shallower bathymetry. This shows the limitations of the linear shallow water equations in shallower regions, although they are computationally more efficient. Therefore in a typical tsunami model to simulate the wave propagation from the deep ocean to near shore it is necessary to consider the nonlinear shallow water equations due to the nonlinear behavior of the wave in the shallower region.

Lack of dispersion in the SWE model is a drawback for tsunami applications, as compared to Boussinesq type models. However, the numerical scheme produces numerical dispersion which is a type of numerical error emerging from the discretization of the governing equations. Therefore, potentially the physical dispersion in the Boussinesq equations can be mimicked by the numerical dispersion generated by the finite difference approximation of the shallow water equations which was first analyzed by Imamura *et al.* (1988) and the method further improved by (Yoon, 2002). The new version of COMCOT is improved and extended and takes into account the weakly nonlinear weakly dispersive waves over a slow varying bathymetry by adding appropriate numerical dispersion in the SWE model (Yoon, 2002; Wang, 2008). It should be mentioned that in order to consider the inherent numerical dispersion based on SWE a specific relation between the grid size and water depth must be fulfilled. This technique is rather easier to be implemented for a 1D model compared to a 2D model. Besides, the anisotropy of the numeric dispersion might constrain the accurate presentation of dispersion. However, some of these problems related to the dispersions are solved and corrected e.g. (Yoon *et al.*, 2007) but difficulties linked to stability and resolution of the grids remain as a challenge. Some tsunami packages based on classical SWE models have the option to take into account the dispersion. For example, in the MOST model the 2D SWE are split into a pair of 1D systems which are solved by a second order accurate finite difference scheme, and the breaking waves are treated as a discontinuity of the solution. The physical dispersion in MOST is captured by numerical terms instead of adding explicit dispersion terms which was extended by Burwell *et al.* (2007). In this respect, the truncation error terms of the Taylor series expansion of the SWE can lead to wave dispersion, though MOST is a shallow water non-dispersive model. In this regard, by proper choice of grid parameters for space and time, the MOST model can reproduce physical dispersive effects. More details of the numerical scheme for implementing the numerical dispersion terms can be found in Burwell *et al.* (2007), and this topic is still an active research area of research in SWE models (Horrillo *et al.*, 2015). Although the MOST

model cannot represent the physical dispersion like Boussinesq type models it is more robust and computationally efficient (Løvholt *et al.*, 2010).

Son *et al.* (2011) introduced a nested modeling of tsunami evolution from generation to propagation. This model couples the Cornell multi-grid tsunami model (COMCOT) to the Boussinesq model presented by Kim *et al.* (2009). The COMCOT model is used for the ocean wave propagation and the final stage of the tsunami including nearshore hydrodynamics and inundation is simulated by a fully nonlinear Boussinesq type model. Coupling two heterogeneous models may produce undesirable results; in this regard a general benchmark test was used to find a parameter range for expected accuracy and stability. Therefore, the potential source of errors due to the physical mismatch between two models may constraint the robustness of these types of hybrid models.

Løvholt *et al.* (2010) developed a simplified version of the Boussinesq model known as global Boussinesq solver (GloBouss) in combination with the MOST tsunami model. The authors argued that the MOST model based on NLSWE is more robust and efficient for the inundation part of the tsunami process. In contrast, existing Boussinesq models might not estimate the run-up process and wave breaking properly, and they are computationally more expensive. MOST and GloBouss models are connected by using one way nesting approach through the boundaries at each time step. Then the boundary values obtained from the GloBouss model are interpolated both in space and time to the MOST model. The authors implemented their proposed model to produce two different types of potential tsunamis generated by landslides and earthquakes. It was shown that the dispersion effect was more crucial for tsunami wave propagation for the landslide generated tsunami case. They also stated their Boussinesq model is not an alternative to the common FUNWAVE and COULWAVE Boussinesq models, but it is a simpler option and computationally much more efficient. GloBouss could also successfully reproduce the 2011 Japan tsunami (Løvholt *et al.*, 2012). The authors demonstrated that the dispersion effect can reduce the leading wave height in the far field compared to traditional shallow water models. It should be noted that dispersion effects are more important for tsunamis with landslide origin because landslides generate shorter wave trains and the frequency dispersion may become important for these tsunamis. On the other hand, SWEs assume hydrostatic pressure, and hence they are not dispersive, which limits their applications for non-seismic tsunamis compared to non-hydrostatic or Boussinesq models (Lynett *et al.*, 2003; Løvholt *et al.*, 2008).

Zhou *et al.* (2012) reproduced the 2009 Samoa tsunami with two different types of tsunami models to investigate dispersive effects and evaluate these models through numerical modelling. First the wave propagation was simulated by a weakly nonlinear and dispersive Boussinesq model based on the (Nwogu, 1993) method. Then under the same conditions, the SWE-based MOST tsunami model with and without dispersion was used to simulate the same event. By comparing the maximum water surface elevations in the domain, it was shown that the non-dispersive SWE predicts higher and steeper wave forms and faster wave speeds. Moreover, after the leading wave excessively dispersive trailing waves were produced by the Boussinesq model which fell beyond the SWE model's capability. The time series of water surface elevation at different gauges from near to far field was also studied. For most of the gauges there were no significant differences between the SWE and the Boussinesq models. However, for some of the far field gauges the Boussinesq model fitted the measurements better than the SWE model which over-predicted the leading peaks. Then the dispersive version of the MOST model was attempted. It was shown that for all cases the MOST model agreed well with the Boussinesq model in predicting the leading wave compared with observations, however, there were some discrepancies for the shorter trailing waves. Overall, the fair agreement with measurements at all stations confirmed the improvement of the dispersive MOST model compared to the non-dispersive SWE model. Finally the authors concluded that regardless of weakly dispersive tsunamis at large scales, the SWE model with numerical dispersion, similar to MOST, may still be an efficient approach to deliver tsunami forecasts in real time.

The finite element method also has been used to model tsunami events from generation to propagation. Behrens *et al.* (2007) developed an unstructured mesh finite element model for tsunami scenarios, known as TsunAWI, to be used for the German-Indonesian tsunami early warning system (GITEWS). Harig *et al.* (2008) used the TsunAWI code to simulate the 2004 Indian Ocean tsunami on different resolution scales from 14 km in the deep ocean to 500 m in the coastal area. The results of the model were compared to available measurements and furthermore evaluated against results achieved by the nested grid TUNAMI-N3 model with a same initial conditions and identical bathymetry/topography focusing in the Aceh region in the northern tip of Sumatra. Both models agreed well in simulating the important features of the first wave including the magnitude, arrival time and inundation area. However, noticeable differences between the two models were found at later stages of the wave propagation. The discrepancy was due to reflections which were influenced by coastal representation in the lower resolved part of the domain. Therefore, it was concluded that the finite element scheme is more suitable because of smooth transition in resolution and consistent treatment of inundation

boundary conditions along the coast, in contrast to the TUNAMI model which uses nested grids with inundation only implemented in the finest grid. In terms of CPU time, the finite element TsunAWI model implemented on a multi-processor shared memory platform by OpenMP parallelism was still 5-10 times slower compared to the finite-difference TUNAMI model. However, the authors argued that their developed early warning system was based on precomputed scenarios, and the CPU time was not the key criterion in their research. It is clear that the finite element model, due to its capability for implementing unstructured grids, is very flexible in discretization of the domain; however, it is computationally very expensive and requires huge computer resources for real time tsunami simulations.

Choi *et al.* (2007) used FLOW3D code to simulate tsunami run-up around a conical island. FLOW3D, solves 3D fully nonlinear dispersive Reynolds-averaged Navier-Stokes equations (RANS) by finite difference method. The results showed a very good comparison with laboratory and 2D numerical results. Although the maximum run-up comparison did not show a significant difference between two models, the flow pattern and velocity distribution were noticeably different. The quantitative RMSE value showed that the non-uniform vertical distribution of the velocity could lead to a maximum difference of about 40% for the velocity field when wave propagated at the lee side of the island compared to the 2D model. Therefore they concluded that the assumption of the depth averaged 2D model is not proper for the conical run-up case.

Kim *et al.* (2013) employed FLOW3D code to study the high run-up observed in the port of Koborinai on the Sanriku coast in the Iwate prefecture caused by the 2011 Japan tsunami. The general numerical model for ocean circulation known as Princeton Ocean Model (POM) was applied to model tsunami wave propagation at basin scale from 2′ resolution, and the results including the water elevation from the boundary domain were interpolated to finer resolutions by a nested grid system up to 10 m resolution provided for the 3D model to simulate the run-up height in the Koborinai region. The important point about the region is the existence of a valley with steep slopes of about 60° where the maximum run-up is observed. The simulation result showed a high vertical speed at about 3 m/s produced by the assumption of non-hydrostatic pressure which is ignored in the 2D model. In this regard, due to the high vertical velocity the simulated flow could climb up to 34.33 m, close to the observed 37.49 m. On the other hand, the 2D POM model with hydrostatic assumption underestimated the wave height at only 25 m. It is evident that the results achieved by 3D models are important for estimation of the tsunami risk for such extremely vulnerable areas. However, it is clear the simulation is

limited to a very specific region which cannot be otherwise reproduced by a single 3D model from ocean scale to such a fine resolution simulation on land.

Abadie *et al.* (2012) studied the numerical modelling of tsunami waves generated by collapse of a volcano in the Canary Islands region. They modelled the initial slide motion by the 3D tsunami model known as THETIS which solves the incompressible, large eddy simulation (LES) filtered Navier-Stokes equations. The initiated waves were then used as the input to the 2D Boussinesq model to simulate the wave travelling towards the nearby islands.

The Smooth Particle Hydrodynamic (SPH) method was first developed by Gingold and Monaghan (1977) to study astrophysical phenomena. This Lagrangian method is used to model many physical problems including astronomy, solid mechanics and fluid mechanics and was recently used to solve SWEs with deforming boundaries and grid distortions (Kao and Chang, 2012; Vacondio *et al.*, 2012). In the context of tsunami modelling the SPH scheme has a superior capability to model breaking details of the wave front and their impact on coastal structures and modelling debris flow following the passage of the tsunami wave (Liu *et al.*, 2013). Rogers and Dalrymple (2008) developed a SPH model to simulate the 2D and 3D cases of tsunami generation and propagation caused by a landslide. The model could reproduce the 2D landslide generated tsunami well, compared to the nonlinear numerical models; however, the number of particles needed for 3D simulation is beyond the capabilities of the regular SPH code requiring high performance parallel computing. The test cases which were used in this paper were at the scale of laboratory experiments. The authors stated that test cases with larger domains required far more discretization than feasible for their presented model. This highlights the current limitations of SPH models in real world tsunami applications. Therefore large-scale tsunami simulation with the SPH model remains inappropriate and impractical.

An extensive review of various numerical models including 3D, SPH, finite difference and finite volume models with application in tsunami simulation and coastal engineering can be found in (Liu *et al.*, 2008).

Due to their superior conservation property and greater flexibility for implementation on various computational grids, high-resolution finite-volume methods have become more popular in recent years for solving the SWEs in tsunami simulations. An example of these finite-volume tsunami models is the GeoClaw software (Berger *et al.*, 2011), which is a modern computing package under constant development that has been widely used for research purposes. In coastal zones, tsunami waves deform in shallow waters and move inland as a surge or wall of water, leading to disastrous inundation in coastal areas. It is ideal to have a model that is able to

represent tsunami propagation in deep oceans as well as inundation in dry lands. Because they enable automatic shock-capturing, finite-volume Godunov-type models have become robust tools in tsunami modelling. They can effectively handle the flow hydrodynamics with the presence of bores and sharp-fronted wave patterns, and can accurately track moving shorelines over complex topographic features. The GeoClaw software based on high-resolution shock capturing finite volume methods has been validated against various tsunami benchmarks (González *et al.*, 2011). By using block adaptive refinement for global tsunami propagation and regional inundation, Geoclaw could reproduce some historical tsunamis such as the 2004 Indian ocean tsunami and 2010 Chilean tsunami (George and LeVeque, 2006; Leveque *et al.*, 2011). The block-structured adaptive mesh refinement framework generates group of grids of different resolution to obtain local refinement and grid adaptation. Melgar and Bock (2013) recently employed the GeoClaw software to study the 2011 Japan tsunami earthquake source model based on joint inversion of near shore wave gauges and land estimates of coseismic deformation from GPS stations.

Another finite volume solver in the context of tsunami simulation was presented by Popinet (2011). In his model the SWEs are solved on adaptive quadtree grids based on a well-balanced positivity-preserving scheme which was proposed by Audusse *et al.* (2004). The model is well-balanced if it can preserve the steady states at rest e.g., the motionless lake at rest. The model proposed by Popinet (2011) could successfully reproduce the 2004 Indian ocean tsunami followed by simulation of the recent 2011 Japan tsunami (Popinet, 2012) which will be reviewed later in this chapter.

In practice, although tsunami propagation in deep water may be accurately modelled at a relatively coarse resolution, it is essential to perform high resolution simulations to reliably reproduce the overland tsunami hydrodynamics in domains with complex topographic features, particularly in coastal cities. This multi-scale requirement for tsunami modelling poses a great challenge for numerical models in terms of improving computational efficiency (Grilli *et al.*, 2007; Ioualalen *et al.*, 2007). In this regard accelerating techniques such as parallelization and adaptive mesh refinement are an advantage which are discussed in the next section.

2.2 Acceleration techniques for tsunami simulations

In this section some of the main approaches to reduce the computational time of the hydrodynamic model are described.

2.2.1 *Mathematical models*

As discussed in the previous section, the models which are often used for tsunami modelling can be categorized as the Reynolds-Averaged Navier-Stokes model RANS (3D), Boussinesq model (2D), and SWE model (2D). Among them, 3D models are known to be the most computationally expensive; however, they can provide detailed information of the flow. Therefore, for practical engineering applications they might be used when the detailed features of the flow are necessary. In the context of tsunami modelling 3D models are more useful for landslide tsunamis e.g., TSUNAMI3D model (Horrillo *et al.*, 2006; Horrillo *et al.*, 2013), or if a very accurate run-up estimation is needed especially for regions which have a steep slope with a complex geometry (Kim *et al.*, 2013).

Boussinesq type models are known as 2D depth integrated models which have been under development for a long time (Madsen and Mei, 1969). In Boussinesq models, the pressure field is not hydrostatic and the horizontal velocities are not vertically uniform. In this respect, there are different types of Boussinesq models which can be derived by implementing a different representative velocity or by considering higher numerical order terms. For example, Nwogu (1993) derived a new form of the Boussinesq equations using the velocity at an arbitrary distance from the still water level as the velocity variable instead of the commonly used depth-averaged velocity. Therefore different models might have different dispersion characteristics. Due to their more detailed mathematical terms which can estimate the physical dispersion of the waves they are used to provide accurate hazard maps in regions which are potentially at tsunami risk. In this regard the FUNWAVE and COULWAVE models are two popular Boussinesq models in the tsunami field. FUNWAVE model and some of its applications discussed in Section 2.1. COULWAVE model is a fully non-linear and dispersive wave model developed by (Lynett *et al.*, 2002a) and it is validated for different applications including multi-layer approach to wave modelling (Lynett and Liu, 2004), wave run-up modelling (Lynett *et al.*, 2002b), and submarine landslide generated waves (Lynett and Liu, 2002). These Boussinesq models are mathematically more complicated and numerically difficult to solve compared to the simpler shallow water equations. Besides, ordinary Boussinesq models are computationally very demanding for large simulations, due to the implicit nature of the solution technique used to deal with dispersion terms, and this issue limits their application for real time tsunami modelling.

On the other hand, the simplest widely used models for tsunami simulations are the linear SWE (LSWE) or the non-linear SWE (NLSWE). The SWEs are derived by the assumption of linear

wave theory in which the wavelength is much larger than the water depth, and viscous effects are ignored. Hence, the pressure is hydrostatic, the horizontal velocity is uniform in water depth and vertical particle acceleration is neglected. These assumptions are usually correct for tsunami waves which are generated by earthquakes. Because the earthquake occurs in a short time and the sea level deforms virtually instantaneously, the wave starts propagating as a gravity wave in the deep ocean. The propagating wave follows linear wave theory until the wave approaches the shallower region when it becomes non-linear. Therefore the entire lifespan of the tsunami process from generation, propagation to inundation and run-up in coastal regions can be represented by the NLSWE. SWE-type models by far are the most popular in tsunami modeling due to their simplicity and efficiency. This justifies the reason why they are used in the most well-known tsunami packages e.g., (MOST, TUNAMI, COMCOT). However, dispersion is neglected in the SWE, and the real shape of the wave during the shoaling process cannot be produced properly although it can still predict the run-up process adequately. Comparison of different methods shows that for practical purposes the NLSWE model can deliver reliable results consistent with its counterparts. Obviously SWE models are the least expensive compared to the 3D and Bousinesq alternatives, while they can provide reliable results. Hence, the SWE are also selected in this research to achieve the most computationally efficient tsunami model.

The aforementioned tsunami models as previously reviewed in this chapter can be solved by different numerical approaches including finite difference, finite volume, finite element and SPH schemes. Among those numerical schemes, finite difference and finite volume methods are the most common in the context of tsunami modelling as they are computationally much more efficient. However, the finite volume methods, as discussed before, are superior due to their conserving and auto shock capturing properties while the solution is also robust.

For more clarification, some of the well-known tsunami models and their characteristics which are discussed in this section are summarized and given in Table 2.1.The computational cost is rating in four different levels, low, medium, high and very high, which these rates are only indicative for a comparison, as illustrated in the last column of Table 2.1. In this regard, numerical model based on SWE is assumed as low computational cost because the computational runtime of the SWE model, might be close to physical time. For example, one hour of wave propagation can be predicted in the scale of minutes. The computational cost for same simulation, using a fully nonlinear Boussinesq model, is few times more (e.g. 10 times more expensive) (Horrillo *et al.*, 2006). 3D models as expected are even more demanding with

one or two order of magnitudes increase in computational time compared to the 2D models (Fringer et al., 2006; Horrillo *et al.*, 2006).

As it is presented in Table 2.1, most tsunami models consider the dispersion. However, tsunami models based on SWE can estimate the dispersion through numerical dispersion and they cannot represent the physical dispersion, e.g. TUNAMI, MOSOT, and COMCOT. Numerical dispersion as discussed previously will not affect the computational time significantly compared to the Boussinesq or non-hydrostatic models which can take into account the physical dispersion such as FUNWAVE and COULWAVE. Horrillo *et al.* (2015) validated and cross-compared eight distinctive tsunami models (including SWE and Boussinesq models) by two laboratory tsunami benchmarks. They concluded all the presented models were adequate for predicting the wave propagation and tsunami run-up for the aforementioned bench mark problems. However, dispersive models were found to perform better than nondispersive models, but differences were relatively small, probably partly due to the solitary waves in the experiments. It would be beneficial, if in future a similar validation and cross-comparison can be carried out for real world tsunami cases in terms of model accuracy and computational efficiency. Also dispersion is not considered in the current study, this is recommended for future work which will be stated in the final chapter.

Tsunami model ID	Mathematical model	Dispersion	Numerical method	Computational cost
MOST	SWE	Numerical	FD	Low
TSUNAMI	SWE	Numerical	FD	Low
COMCOT	SWE	Numerical	FD	Low
GeoClaw	SWE	No	FV	Low
NEOWAVE	Non-hydrostatic SWE	Physical	FD	Medium
GloBouss	Boussinesq	Physical	FD	Medium
COULAWAVE	Boussinesq	Physical	FD+FV	High
FUNWAVE	Boussinesq	Physical	FD+FV	High
TsunAWI	SWE	No	FE	High
THETIS	3D	Physical	FD	Very high
FLOW3D	3D	Physical	FD	Very high
TSUNAMI3D	3D	Physical	FV	Very high

Table 2.1 Summary of model characteristics.

In terms of numerical schemes, as discussed in this section, in general, tsunami models which are solved by conventional finite difference (FD) schemes are computationally more efficient

than SPH and finite element (FE) approaches which are computationally more demanding. Computational performance of finite volume scheme (FV) is dependent on the grid type, as finite volume solver scan be implemented on structured and unstructured grids. Unstructured grids can provide more flexibility for complex geometries than structured grids. While the computational burden of finite volume solver on structured grids is reduced, because no data structure is necessary for relating cells, and trigonometric functions are not required to update cells. The computational time of finite volume scheme on structured grids is comparable to finite difference method; hence, a finite volume scheme on structured grids is considered in this study.

The proposed hydrodynamic model in this work solves the 2D SWE using a finite volume Godunov-type scheme incorporated with Harten, Lax, and van Leer with Contact wave restored (HLLC) approximate Riemann solver. Effective numerical techniques are implemented to ensure a well-balanced solution of the lake at rest problem and to accurately track moving wet-dry shorelines (Liang, 2010).

2.2.2 *Adaptive grid techniques*

An efficient way to improve the computational efficiency of a hydrodynamic model is to use adaptive mesh refinement (AMR) techniques, which provide an effective means to support multi-scale tsunami modeling. The tsunami wave propagating in the deep ocean follows the linear wave theory as the wavelength is large, and coarse resolution is sufficient to simulate the wave evolution from the source to near shore region. However, for evaluating the tsunami wave in the coastal area, accurate high-resolution bathymetric and topographic data are crucial, with the maximum grid size recommended to be 2 arc second (~ 60 m) (Wei *et al.*, 2013) and fine grids must be used for reliable simulations.

Major conventional tsunami models solve the SWEs by a finite difference scheme as discussed earlier in this chapter. Usually in conjunction with a finite difference scheme, the nested grid technique is adopted to consider various resolutions in the whole model, as implemented in the TUNAMI, COMCOT and MOST models. For example, the MOST tsunami model employs nested grids to telescope down to the area of interest for computing the inundation where high resolution grids are required. In order to minimize the error in the solution of the equations, the optimum number of computational nodes per wavelength are generated by nested grids. Shuto (1991) reported that, according to numerical experiments, one local tsunami wave length should be covered by over than 20 nodes in order to eliminate the numerical decay in wave height. Then these nested grids are coupled in one direction from the outer grid. The outer grid is not

affected by the inner grid, and the outer grid provides the inner grid with calculated wave heights and velocities at all four directions by linear interpolation from coarser resolution. The MOST model which employs nested grids has been validated for the inundation phase for historical tsunamis, e.g. (Synolakis *et al.*, 2008; Tang *et al.*, 2012; Zhou *et al.*, 2012).

In the COMCOT tsunami model, the SWE on either a spherical or Cartesian system can be assigned to a specific sub-level region, and these sublevels are dynamically connected. Any integer ratio of the grid size between two directly nested grid layers is possible. The details of this nested grid technique are reported in (Wang, 2008).

An alternative method to static nesting grids are dynamic adaptive techniques where the resolution during the simulation adapts to follow features in the flow. The common practice is to implement dynamic embedding of Cartesian grids, e.g. (Berger and Oliger, 1984) which has been successfully applied for tsunami modelling in GeoClaw software (George and LeVeque, 2008; Berger *et al.*, 2011) and more recently used to simulate the 2011 Japan tsunami (Arcos and LeVeque, 2015). Another well-known AMR is the hierarchical method e.g. (Rogers *et al.*, 2001; Liang and Borthwick, 2009) which also has been used for tsunami simulations (Popinet, 2011; Popinet, 2012). As these techniques implement local mesh refinements by employing data structures with various levels of complexity, they require computational overheads to save and search for these data structures. Therefore to develop a method which can minimize the overheads to manipulate local mesh refinements might be an advantage. In this regard Ji *et al.* (2010) developed a new cell based structured AMR data structure to improve the efficiency of the conventional data structure of the quad-tree technique; but this technique has not been applied in the context of tsunami modelling.

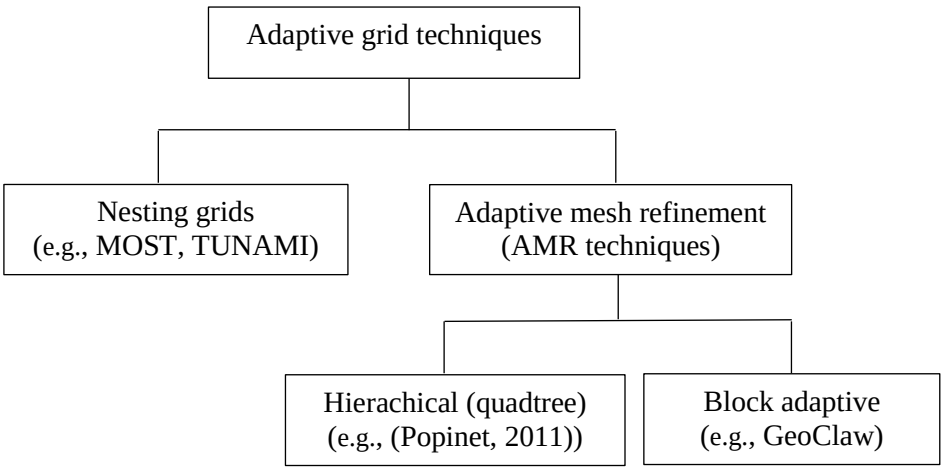


Figure 2.2 Adaptive grid techniques diagram for tsunami simulations.

Although AMR techniques have their advantages, the problem in maintaining mass conservation and well-balanced conditions might limit the development of this type of approach for a broader range of applications (Popinet, 2011). The adaptive grid techniques which have been used for tsunami simulations and briefly reviewed in in this subsection are illustrated as a diagram in Figure 2.2.

Liang (2012) proposed a simplified AMR method with no data structure needed, and recently used this technique to simulate tsunami wave propagation (Liang *et al.*, 2015a). The details of this method will be explained in Chapter 6.

2.2.3 *High-performance computing*

- Parallel computing techniques

With the advancement in computer technologies and available resources, high-performance computing is becoming more popular as it can deliver higher performance through different computing architectures. Also to model real world applications which deal with large scale domains high-performance computing might become necessary. In this regard parallel computing techniques can save computational time by executing the independent parts of a program simultaneously on different processors. The parallelism can be implemented on distributed or shared memory architecture.

In distributed memory architecture, each processor has its own private memory and they can be connected to other processors by networks. Different parts of the whole task are assigned to different processors in a cluster. Thus, the computational domain needs to be decomposed into subdomains, and each domain can be then executed by a separate processor. The memory is scalable with the number of processors; however, the data communication between processors is passed through message passing interface libraries (MPI). For example in the finite volume Godunov type scheme the cell information and its neighboring cells are required at each time step. Clearly, due to domain decomposition, the information interchange between the adjacent boundaries of each subdomain is necessary. Therefore, this technique is constrained by communication between neighboring cells, and scalable implementation is relatively difficult to accomplish (Pau and Sanders, 2006; Delis and Mathioudakis, 2009).

The alternative parallel computing technique is based on the shared memory architecture, in which multiple processors have access to the global memory. The parallel version of a sequential program on CPUs is carried out using the OpenMP programming interface. For example Neal *et al.* (2011) developed the parallel version of the LISFLOOD hydraulic model

successfully, and they showed the model speed-up through different resolution and domain sizes. The main advantage of shared memory parallelization on CPUs is related to its simplicity as it does not require much programming effort or changes in the original code. However, the lack of scalability between processors and memory is the main disadvantage.

A more promising high-performance computing technique involves the use of graphics processing units (GPUs) and has started to gain rapid popularity in the last few years. GPUs have been commonly used in the game industry but are only recently available for scientific computing (Brodtkorb, 2010; Brodtkorb *et al.*, 2010). There are hundreds of processing elements on a single GPU to provide powerful parallel computing capability, in contrast to a central processing unit (CPU). The benefit of using GPUs to provide high-performance computing is evident. In less than one decade, numerous GPU-accelerated models have been developed and used in many areas of scientific computing, e.g. computational fluid dynamics (CFD), magneto-hydrodynamics, and gas dynamics (Wang *et al.*, 2010; Kuo *et al.*, 2011; Rossinelli *et al.*, 2011; Schive *et al.*, 2012). In order to use the GPU technology for a parallelizable program, the original sequential CPU code must be updated to be compatible with the GPU configuration. This update requires effort depending on the complexity of the problem but the performance gain justifies the GPU implementation.

In the next part these parallel computing techniques are briefly reviewed in the context of SWE modelling using different numerical techniques.

2.3 Review of parallel shallow flow models

Boussinesq wave models which are used for tsunami modelling are usually computationally very expensive, consuming huge amounts of computer resources including CPU time and memory due to the implicit nature of the solution technique employed to consider the dispersion terms. Therefore, for large scale problems, it is either not affordable or feasible to run them on a single PC. Through parallel computing the fully non-linear Boussinesq models can be implemented on large clusters by employing nested grid algorithms, e.g. FUNWAVE tsunami model parallelized by Shi *et al.* (2012), and the model has been used to simulate the recent 2011 Tohoku tsunami event using fine resolution grids (Grilli *et al.*, 2012). Sitanggang and Lynett (2005) developed the parallel Boussinesq model of Wei *et al.* (1995) using the domain decomposition technique in conjunction with a message passing interface (MPI) to facilitate such intensive computations. The high order finite difference part is parallelized similarly to its serial version, and the tridiagonal matrix systems are parallelized by implementing a parallel matrix solver integrated with an efficient data transfer scheme. Adopting a similar approach,

Pophet *et al.* (2011) explored the use of multi-core parallel computing to achieve improved computational efficiency for the FUNWAVE tsunami model solving the Boussinesq equations. A similar parallel algorithm is also used by Delis and Mathioudakis (2009) to develop their shock-capturing finite volume model that solves the SWEs.

Son *et al.* (2011) parallelized the COMCOT model in their hybrid model in conjunction with the Boussinesq model for implementing on shared memory computers such as multi-processor or multi-core computers. In this regard OpenMP was utilized for the parallelization, as a standard method for shared memory parallelization. The model was tested for 8 processors, which showed a near linear speed up reducing the computational time by a factor of 8.

Oishi *et al.* (2015) parallelized the TUNAMI-N2 model for tsunami inundation forecasting with application to the 2011 Japan tsunami. They developed the model to be executed on the 'k' supercomputer known as Japan's fastest and the world's fourth fastest computer. The model is parallelized by a 1D or 2D domain decomposition method. It was discussed that the load balancing between nested domains and the number of subdomains in each nested domain are the factors to achieve good performance. By employing 9,469 CPU cores, the model was able to predict two hours of inundation using 5 m resolution and a 0.1 s time step in about 1.5 min (75 times faster than real time) in Sendai city for the 2011 Tohoku tsunami. Such a significant performance of the model can provide accurate estimations of tsunami behavior in coastal areas through real time simulations with a proper resolution in combination with source prediction techniques. However, such a performance as presented in their work can be only achieved by using a supercomputer or computer clusters which require huge investments and are not available for individual researchers.

In computational hydraulics that focuses on SWE models, Brodtkorb (2010) implemented a Kurganov-Levy and Kurganov-Petrova numerical scheme to solve the SWEs on a GPU and test the CUDA (Compute Unified Device Architecture) based heterogeneous architectures for improved computational performance. More recently, Smith and Liang (2013) presented a second-order accurate finite volume Godunov-type SWE model on GPUs. Due to the use of OpenCL programming framework, their model can be run on any modern GPUs and CPUs and therefore offers greater flexibility in model applications. Both of these GPU SWE models were developed and tested for flood modeling and showed a significant performance increase of an order of magnitude compared to the same model executed on a single CPU.

In terms of application for flood simulation Vacondio *et al.* (2014) developed the first order and second order shallow water numerical scheme by using the CUDA framework available for

NVIDIA GPUs. They implemented some optimization procedures such as block deactivation optimization, the implicit approach, and attempted to minimize the information exchange between CPU and GPU which delivered a computational performance improvement of an order of magnitude in comparison with serial code. The algorithm which was used for the numerical scheme showed that second order accuracy in space and time is about three times slower that the first order accuracy. Application of this model to the 2009 Parma river flood event considering second order accuracy showed that the acceleration ratio of physical time to computational time is about 18 times. Also the performance of different GPU devices including GTS, GTX and Tesla GPUs was tested experimentally. It was concluded that a regular cheaper GTX GPU with a greater number of cores and clock speed could deliver better performance in comparison to the more advanced Tesla M2070 GPU, whereas the latter device is the only option for real case simulations with larger grid sizes due to their large memory.

Due to the design of GPU devices, structured grids are more suitable than unstructured meshes. However, Lacasta *et al*. (2014) introduced some GPU optimization strategies related to cell and edge ordering for their finite volume scheme on unstructured grids to improve the computational gain. Then the improved GPU solver on an unstructured grid could speed up the simulation by an order of magnitude compared to the single CPU; however, the simulations were only performed for two classical benchmarks and not for a real application. In their recent work, they extended their shallow flow model to consider the bed load transport via an efficient GPU implementation (Juez *et al.*, 2016). They applied the model to simulate a laboratory test case and a real dam break. They showed that their GPU model could provide a peak speed-up of 50 compared to the single core CPU.

As the SPH shallow flow models are computationally very demanding by nature, intensive research has been carried out in terms of high-performance computing. More recently, the application of GPU technology is highlighted and research is ongoing (Crespo *et al.*, 2011; Domínguez *et al.*, 2013; Valdez-Balderas *et al.*, 2013; Nishiura *et al.*, 2015; Xia and Liang, 2016). Nishiura *et al.* (2015) evaluated the performance of SPH on several shared memory parallel computer architectures including GPUs and multi-core CPUs. They found that in their work the GPU could produce the best arithmetic performance and deliver high computational speed while it gives the most efficient power consumption. In a different study Domínguez *et al.* (2013) presented the implementation of multi-GPU for SPH on heterogeneous clusters including CPUs and GPUs. The communication between devices was facilitated by an improved MPI implementation which addresses some disadvantages of the common issues of MPI including dynamic load balancing and overlapping data communications. Therefore, by

combining several GPUs and the efficient use of memory, this enabled simulations with large number of particles. In this regard they could model a realistic interaction of a large wave with an oil rig in a supercomputing centre, considering more than 10^9 particles. The run time for 12 s of physical time was about 92 hours using 64 GPUs M2090.

Gandham *et al.* (2015) presented a GPU accelerated discontinuous Galerkin (DG) method for solving the shallow water equations. The performance of the kernels was compared in a portable threading language, when cross compiled with OpenCL, CUDA, and OpenMP. The model was validated through some classical benchmarks, and the experiments carried out on an NVIDIA Titan GPU. It was shown that better performance was achieved by CUDA compared to OpenCL, as the CUDA compiler can be optimized to its hardware, reporting the same conclusion as in (Danalis *et al.*, 2010).

A GPU based simulation of a tsunami was introduced by Liang *et al.* (2009) at the early stages of developing scientific codes on GPUs. In this model, a second order finite difference MacCormack scheme was used to solve the NSWE. They utilized the parallelism by designing algorithms in shaders using OpenGL shading language. The simulations were carried out on a laptop with a GeForce 8400m GPU to simulate a test case with 80 × 180 nodes in the domain. It was shown that the speedup of the GPU implementation over the CPU version was about two times on average. However, for their future work they suggested to implement the new computing technology CUDA, in order to get more control in parallelizing applications to achieve better performance.

Gidra *et al.* (2011) presented a high performance tsunami prediction based on the linear TUNAMI-N1 model on constant grids using GPU technology in the CUDA framework. They proposed two parallel models known as the data parallel and hybrid model. In their data parallel model the wave height of the tsunami at each point of the bathymetry file is computed in parallel. In the hybrid model, independent functions are calculated at different blocks, where each functional unit computes the output value in parallel for every point in the bathymetry file. They also implemented optimization and memory management techniques to improve the efficiency. The experiments were carried out on Quadro FX 1700 GPU which only supports single precision floating point causing a mild numerical inaccuracy in the final results. Finally, it was shown that the parallel model could speed up about six times compared to the serial version of the original Fortran code.

In terms of high-performance tsunami wave propagation modeling Lavrentiev-jr *et al.* (2009) sped up the MOST model as an initial propagation software engine accepted by the USA NOAA

as the basic tool to compute tsunami wave propagation for a shared memory system (OpenMP), with distributed memory systems using the message passing interface (MPI) and CELL architecture. Then recently they developed and extended their parallel platforms to include GPUs which showed a speedup of about 40 times for the 2011 Japan tsunami (Vazhenin *et al.*, 2013).

In another study the GeoClaw software was parallelized by using OpenMP to study the near-field Tohoku tsunami at a high grid resolution (Zhang *et al.*, 2011; Song *et al.*, 2012). They could achieve over 80% of the potential speed-up on a multi-core system. They carried out the simulation of 40 minutes tsunami propagation from a source in the deep ocean to the nuclear power plant of Fukushima with a 4^{th} level of refinement and 20 m grid spacing. The computation took more than two hours of wall clock time with 16 threads on an eight core Dell Precision T7500 work station; it would have taken over 16 hours on a serial version of the code. The authors discussed that the OpenMP GeoClaw is becoming more popular as nowadays almost all laptops and desktops are equipped with multi-core architectures, also more multi-core systems (SMP machines) have become available. It was argued that domain decomposition by the MPI parallel technique has not been shown to be an efficient method to deal with adaptive mesh refinement (AMR) in the Geoclaw software. Furthermore, the authors mentioned that outstanding performance has been reported for GPU implementation in the high-performance field, but it might take some years to implement GPU for the GeoClaw AMR software to be capable of solving real world applications. Therefore the authors concluded the Geoclaw software could scale more efficiently and feasibly on many–core systems. In similar research in the context of high-performance computing, Breuer and Bader (2012) used the GeoClaw code to study parallelization with CUDA, MPI, OpenMP, and hybrid approaches mainly for teaching purposes in computational science and engineering applications. The 2011 Tohoku tsunami was selected as test case with a uniform grid of 14000 × 8000 cells with a resolution of about 500 m. The model could scale up nearly linearly in the case of MPI parallelization using different types of numerical solvers. In terms of hybrid parallelization (OpenMP+MPI) it was shown that using one MPI process with 8 OpenMP threads was 20% more efficient compared to using 8 single-thread MPI processes. Finally the results for a small scale simulation of 800 × 800 cells by implementing GPU were investigated. However, the speed up of the GPU model was not presented, and the authors mentioned that in the near future they intend to employ the software on GPU clusters (MPI+CUDA) as part of their parallel programming models.

Acuña and Aoki (2009) used the TSUBAME GPU cluster for the real time tsunami simulation on multi-node GPU cluster. They achieved a perfect scaling on a cluster of 32 GPU nodes, by overlapping computations and communication. Their single node GPU implementation, to solve the SWEs by a conservative semi Lagrangian method, is approximately 62 times faster than corresponding CPU version, and they estimate that a simulation of a 4096×8192 mesh with 90m resolution, will run within 3 minutes by using 8 GPUs. In terms of GPU-CPU scalability they estimated that 1024 CPUs are required to match the performance of 16 GPUs. However, in their work the performance of the model was only shown for an initial wave resembling the properties of a fictitious tsunami by two test bathymetries. Therefore, for practical applications, further work is required to evaluate the accuracy and performance of the model against a real-world tsunami case.

Castro *et al.* (2010) investigated the implementation of high order finite volume schemes to simulate shallow flows on GPUs and discussed the advantages of using GPUs for finite volume schemes. They stated that finite volume scheme performs a high degree of data parallelism as the calculation at each computational cell is independent with respect to the calculation performed at the rest of the cells. Besides, the finite volume scheme shows a high arithmetic intensity. They carried out their simulation for a circular dam-break problem with a NVIDIA GTX 280 GPU. They found that for large size uniform problems, the GPU implementation by using single precision could speed up the simulation for two order of magnitudes than the CPU version. However, the model using double precision reported to be 7 times slower than the single precision which was due to the limited number of double precision units on GTX 280 GPU compared to the graphics cards with FERMI architecture. The execution times using two different languages for GPU parallel programming including CUDA programming framework and shading language (Cg) were also compared. In this regard, it was shown that the CUDA program outperformed with respect to a previously GPU shading language. Recently, the authors extended their model for a two-layer shallow flow model to simulate tsunamis generated by landslides on multiple GPUs (de la Asunción *et al.*, 2016). In their recent work they developed the load balancing algorithm to deal with the variations of the wet and dry regions between the subdomains which can improve the speed up compared to one GPU up to a 12% in real problems. The results show good scalability for several test cases by using up to 24 GPUs with respect to the equitable computational load among the subdomains.

Wang *et al.* (2010) were probably the first who mapped an adaptive mesh refinement on GPU, with the application in magneto hydrodynamics. The authors used the implementation of Berger and Colella (1989) block structured AMR framework to the open-source astrophysics code

ENZO, the recent version of ENZO is described in Bryan *et al.* (2014). Wang *et al.* (2010) considered the mapping of hydrodynamic solver onto GPU while they processed the grid hierarchy and creation on CPU. All problems were executed on a Quadro FX 5600 GPU, and the model showed an overall speed up of about 10 times runtime on GPU as compared to a single CPU. Although, ENZO considers non-uniform patches, but it does not exploit concurrency between the CPU and GPU, and involves a large number of GPU kernel invocations. Raghavan and Vadhiyar (2015) developed strategies for adaptive execution of AMR applications on GPUs including adaptive asynchronism between CPU and GPU executions. They showed that their model with asynchronous strategies and kernel optimizations could improve the computational performance by at least 1.98 times compared to the ENZO model.

Schive *et al.* (2010) developed GPU accelerated adaptive mesh refinement code (GAMER) for astrophysical simulations based on a hierarchy of grid patches with an oct-tree data structure, in which each patch is restricted to maintain a constant number of cells. In this work by employing the capability of asynchronous memory copies in GPU, the computational overhead regarding to data transfer between the CPU and GPU is reduced. It was shown that the GAMER code for cosmological simulations by using one GPU could speed up the execution time about 12 times compared to the CPU version of the code. The authors improved the GAMER by implementing several optimizations in the code, including the asynchronous memory copy, the concurrent execution between CPU and GPU, and the hybrid MPI/OpenMP/GPU parallelization, for fully exploitation of the computing power in a heterogeneous system (Schive *et al.*, 2012). GAMER model by using fixed patches at the same size avoids the problem with load balancing which will affect the computational performance. Because, uniform patches can lead to over-refinement which will consequently increase the computational costs (Raghavan and Vadhiyar, 2015).

The new generation of GPUs with their advanced architecture, e.g. Kepler GPUs, are capable of dynamic parallelism without CPU intervention. Therefore, by using modern GPUs, easier implementations of AMR techniques on GPUs are expected, which will be discussed in more details in Chapter 6-Section 6.3. In the context of tsunami simulations, the implementation of AMR techniques for finite volume schemes on GPUs has not yet been reported in the literature. This thesis introduces a static mesh refinement on GPU for tsunami modelling which will be demonstrated in Chapter 6-Section 6.4.

The parallel computing techniques for tsunami simulations which have been discussed in this section are summarized and shown in Figure 2.3, which shows the common parallel techniques including CPU, GPU and hybrid parallelism.

The number of GPU based hydrodynamic models is increasing; however, few GPU-accelerated models have been reported for tsunami simulations. This work aims to implement GPU technology by using the CUDA framework for the proposed hydrodynamic model.

<table>
<tr><td colspan="3">Parallel computing</td></tr>
<tr><td colspan="2">CPU parallelism</td><td>GPU parallelism</td></tr>
<tr><td>Shared memory
(OpenMP)
e.g. GeoClaw
(Zhang et al., 2011)</td><td>Distributed memory
(MPI)
e.g. FUNWAVE
(Pophet et al., 2011)</td><td>Shared memory
(CUDA)
e.g. MOST
(Vazhenin et al., 2013)</td></tr>
<tr><td colspan="3">Hybrid parellelistion
(OpenMP+MPI) e.g. GeoClaw (Breuer and Bader, 2012),
(CUDA+MPI) e.g. (de la Asunción et al., 2016)</td></tr>
</table>

Figure 2.3 Parallel computing techniques for tsunami simulations.

2.4 Source models

An intensive research has been conducted to study the tsunami source especially after the 2011 Tohoku-Oki earthquake. The investigation has involved different approaches by using tsunami wave data, teleseismic data, GPS data or combination of the aforementioned methods (Ammon *et al.*, 2011; Fujii *et al.*, 2011; Hayes, 2011; Ozawa *et al.*, 2011; Pollitz *et al.*, 2011; Shao *et al.*, 2011; Yamazaki *et al.*, 2011b; Gusman *et al.*, 2012; Iinuma *et al.*, 2012; Ohta *et al.*, 2012). Some of these earthquake slip models from the literature are summarized and presented in Table 2.2.

The conventional approach based on seismic inversions is a rapid method to estimate the source model in a real time forecasting scenario as the seismic waves travel much faster than the gravity waves. To determine the seismic inversion different methods can be used. For example the detailed spatial dimensions and slip distribution of the source can be constrained by the finite fault inversion algorithms. This method can characterize the earthquake source rapidly as developed and discussed in detail in (Ji *et al.*, 2002; Hayes, 2011). This method of source determination has been applied by other researchers to use as the input for their tsunami models, e.g. (Wei *et al.*, 2013; Yamazaki *et al.*, 2013).

For the 2011 Japan tsunami source the deformation model obtained by Shao *et al.* (2011), United States Geological Survey (USGS), is known to be even more precise and the results are compatible with tsunami wave measurements as reported in (Popinet, 2012). The algorithm which was developed by Shao *et al.* (2011) monitors large earthquakes by jointly inverting body and surface waves. In their work, unlike conventional moment tensor inversion methods, an earthquake source is considered as multiple double-couples to determine focal mechanism variation throughout the coseismic rupture of major earthquakes. The authors mentioned their solution estimated the slip to be 50-60 m, about two times more than previous reported solutions which was surprising. Similar large slip near the trench was also reported by other researchers using an alternative teleseismic approach (Lay *et al.*, 2011a).

Model ID	Data type
Fujii *et al.* (2011)	Tsunami
Satake *et al.* (2013)	Tsunami
Shao *et al.* (2011)	Teleseismic
Hayes (2011)	Teleseismic
Iinuma *et al.* (2012)	GPS
Yamazaki *et al.* (2011b)	Teleseismic and tsunami
Ammon *et al.* (2011)	Teleseismic and GPS
Gusman *et al.* (2012)	Tsunami and GPS

Table 2.2 Inversion-based slip models from the literature.

Estimating the source model from tsunami wave data is also an important approach which can provide useful information about the tsunami propagation from the source to near shore region. In this context the work of Fujii *et al.* (2011) is noteworthy. They estimated the slip distribution from different types of gauges including bottom pressure gauges, offshore GPS gauges and 33 coastal tide gauges of the 2011 Tohoku earthquake. Then they expanded their research by utilizing 53 gauges, also they considered both time and space distribution of coseismic slip as deduced from tsunami waveform data (Satake *et al.*, 2013). They showed that the delayed rupture in their model could produce better fit to the tsunami waveforms observations compared to the instantaneous rupture, especially for the stations located in the northern of the tsunami source origin.

The information obtained by deep ocean tsunami measurements known as tsunameters (DART buoys) can also predict the tsunami source accurately. Because the inversion process is based on the assumption that tsunami propagating at locations of these gauges is linear so that the waves initiated by uplift caused by slip on subfaults can be linearly superimposed. On the other hand, when the wave reaches the shallow region it will shoal and the assumption of linearity might not be correct any more. Therefore, the information from tsunameters in deep ocean can be a reliable option while the wave propagation follows a linear process and is not affected by the local bathymetry or other interactions from the coast.

Considering different technology, the speed and accuracy of tsunami warnings can be increased by employing GPS instruments. The prediction time can be decreased by 20 minutes which can potentially reduce the damage to coastal communities according to recent research (Melgar *et al.*, 2016). However, the importance of this technology for tsunami early warning systems has had the attention of researchers in this field for over a decade, especially after 2004 Indian ocean tsunami. GPS stations can provide more accurate information related to the ground displacements compared to seismological stations. Because seismological stations are more suitable to investigate the ground shaking caused by earthquake rather than ground offsets. However, both systems have their own advantages and should be considered together. The 2004 Sumatra earthquake (M_w ~ 9.2-9.3) triggered the deadliest tsunami in history, however, the real magnitude of the earthquake was not determined by seismic estimates, and was significantly underestimated (M_w ~ 8.0-8.5). This issue arises from the inherent saturation of early seismic approaches. Analysis based on the earth's longest period normal modes confirmed larger magnitude close to reality in few days later (Park *et al.*, 2005; Stein and Okal, 2005). Therefore an intensive research has been carried out for rapid determination of earthquake magnitude using new technologies for tsunami warning systems.

In this regard Blewitt *et al.* (2006) showed that GPS data measuring the average displacement of the land's surface associated with the arrival of seismic waves can determine the earthquake's real size and tsunami potential within 15 minutes after the earthquake generation. They also stated that their results concerning the magnitude of great earthquakes are highly sensitive to the number of GPS stations in the near field which means the presence of real-time GPS networks near oceanic subduction zones is an advantage. The seismic data are also crucial for fast detection and position of potentially important events. Then GPS data could be implemented for fast earthquake modelling and consequently determine parameters for real time simulation of tsunami initialization. Moreover, by employing ocean sensor data, the tsunami model can be further constrained and validated. Hence, seismic, GPS, and oceanic data

can be integrated in tsunami warning systems which have been continuously under development (Titov *et al.*, 2005).

For example, Ohta *et al.* (2012) developed an algorithm to estimate static ground displacements due to earthquake faulting from real-time kinematic GPS time series. By inverting the displacement for a fault model, they estimated a fault with M_w 8.7 magnitude, close to the real value M_w 9.0, in less than five minutes from the elapsed time of the 2011 Tohoku earthquake. They mentioned that the coseismic seafloor deformation near the trench which was about 200 km away from the coast was difficult to be estimated properly by onshore GPS data. Therefore, the tsunami source achieved by GPS data can be improved by adding offshore tsunami data which represent more information from the actual tsunami source. By improving the tsunami source, the coastal gauges then can predict the tsunami wave more accurately. In this regard, the authors applied the algorithm proposed by Tsushima *et al.* (2009) to their source fault model and added the wave information at the offshore buoys 15 minutes or later after the earthquake. They showed that the improved source model could reproduce more reliable predictions for the 2011 Tohoku tsunami. Finally they suggested to use the inland real-time GPS data along with seismological approaches because these methods are complementary which will lead to more accurate and robust tsunami forecasting for the next mega earthquake.

Each of the aforementioned source models as discussed so far has its own merits and limitations. Wei *et al.* (2014) used three different types of tsunami sources including (tsunameter, tsunameter/GPS, and finite-fault) as the input to the MOST tsunami model. They showed that the tsunameter derived-source model on average could reproduce more accurate wave propagation compared to observations at six GPS buoys rather than the two other source models.

In this project, two different fault models from the literature are adopted as the source models for the 2011 Japan tsunami. These source models are based on tsunami wave form and tsunameter inversions, as reported by Fujii *et al.* (2011) and Wei *et al.* (2014) respectively. Each source model provided the parameters of different sub faults. The vertical sea floor deformation will be computed by employing the elastic model of Okada (1985) which is implemented by Clarke *et al.* (1997). Instantaneous rupture is assumed because the total rupture time of 2011 Tohoku earthquake is short enough (3 minutes) to be negligible. More details will be explained in Chapter 5.

So far different tsunami models, numerical schemes, source models and acceleration techniques have been reviewed. Figure 2.4 demonstrates an example of a typical integrated model for tsunami simulations from tsunami generation to tsunami inundation in a diagram.

The next section reviews implementation of some of these methods with a focus on the 2011 Japan tsunami.

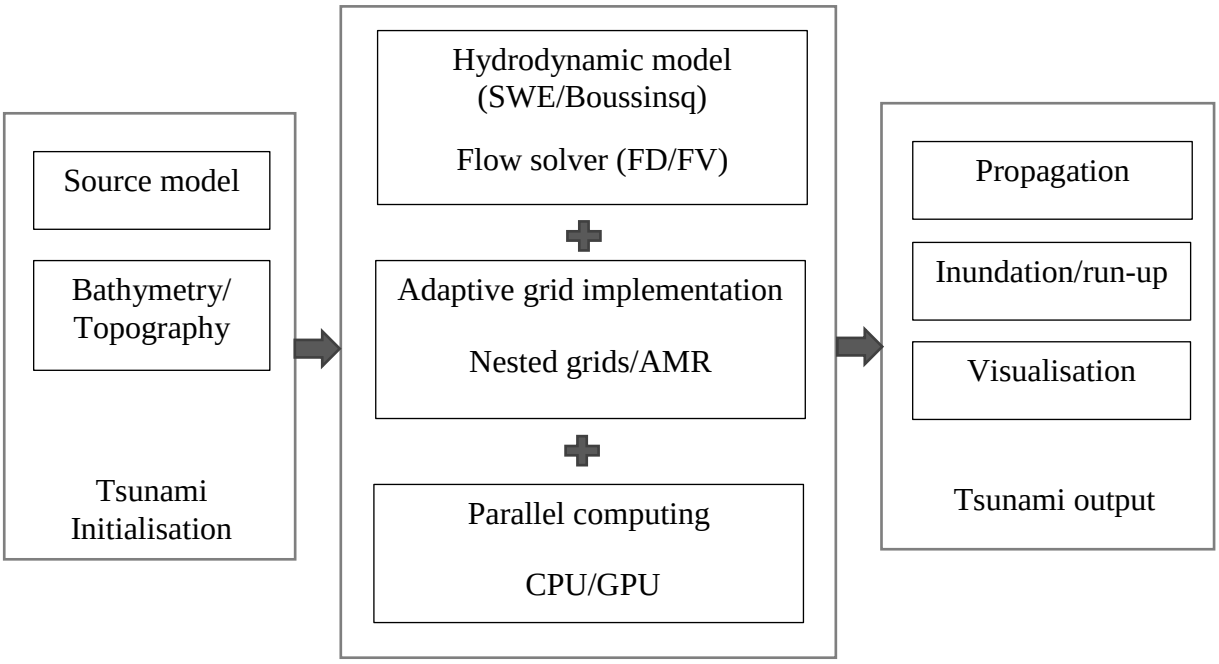


Figure 2.4 Diagram for tsunami simulation from generation to inundation.

2.5 Review of 2011 Japan tsunami simulations

Yamazaki *et al.* (2011b) used a non-hydrostatic pressure term in the non-linear SWEs which also includes a vertical momentum equation known as NEOWAVE (Non-hydrostatic Evolution of Ocean Wave) to describe tsunami generation from sea floor deformation and propagation of weakly dispersive tsunami waves. In the aforementioned work a modification to initialization is implemented which improves the fit to the various tsunami data while keeping the fit to the seismic and geodetic observations. The rupture models provide the finite-fault slip time histories to prescribe kinematic deformation with the planar fault model of (Okada, 1985). The uncertainties in the model parameters lead to many modelling approximations.Yamazaki *et al.* (2011b) showed the proper changes in strike and dip parameters and reduction can provide a better fit compared to the observations. For example they found that by reducing 5–10° of average fault strike, and slight steepening of average fault dip the simulations can be improved compared to the near-field observations. The uncertainties in the model parameters lead to many modelling approximations. The numerical results for the deep Ocean Dart21418 from (Lay *et al.*, 2011b) and (Yamazaki *et al.*, 2011b) show that modification of the initial condition has a significant impact on the prediction as both use the identical hydrodynamic model. However, apart from the sensitivity of the model due to initialization, there is no discussion for the

proposed hydrodynamic model compared to SWEs which only consider hydrostatic assumptions.

Goda *et al.* (2014) carried out an intensive sensitivity analysis for the tsunami wave profiles of the 2011 Tohoku earthquake by focusing on the earthquake slip and fault geometry by considering stochastic slip distributions and inversion based slip models based on the spectral analysis approach of Mai and Beroza (2002). It was concluded that the source model is highly sensitive to site location, the characteristics of the slip, and also to various dip values. While the dip variations in the subduction zone are relatively well known and with depth or along strike may alter gently, the major source of uncertainty for predicting tsunami hazards to future events is highlighted as the earthquake slip characteristics.

In terms of slip distribution Yamazaki *et al.* (2013) stated that the large slip near the trench produced a long-crested wave propagating toward the shelf. The superposition of the long-crested wave with the radiated wave generated from the epicenter region produced a high amplitude of 6 m which is recorded near-coast. The high amplitude wave with local topographic effects can cause large run-up heights as occurred in the Iwate coast.

As reported by Maeda *et al.* (2011), two Ocean bottom pressure gauges (OBP) called TM1 and TM2 off Kamaishi, Northern Japan, as shown in Figure 2.5, recorded the 2011 tsunami. It was recorded that the wave height before reaching the coast was more than 5m. The observation shows a gradual rise from 0 to 2 m during the first 800 s from the earthquake elapse time, followed by sharp uplift of tsunami wave to 5 m in 200 s. The characteristics of these waves show that the largest slip occurs in the shallow portion of the plate interface near the trench axis, and smaller slip is generated in the northern part near the coast. Such characteristics of tsunami waves shaped above the source cannot be recorded in the tide gauges near the coast because the tsunami wave stages from gradual to sharper rise observed in the OBP gauges when reaching the coast will coincide to produce one large wave with lower propagation speed. The large slip predicted near the trench is consistent with other studies derived from teleseismic waveform data e.g. (Ammon *et al.*, 2011; Ide *et al.*, 2011; Shao *et al.*, 2011).

Large horizontal displacements measured near the trench axis, about 50-70 m in addition to the vertical deformation are also inferred (Fujiwara *et al.*, 2011; Ito *et al.*, 2011). The large horizontal displacements affect the tsunami generation if the sea floor is steep. This horizontal displacement near the trench can increase the tsunami amplitude by 30% as shown by (Tanioka and Satake, 1996). It was shown by Satake *et al.* (2013) the computed tsunami waveforms

including the horizontal component are about 30-60% larger compared to those from the vertical component only.

Satake *et al.* (2013) showed that the tsunami wave predicted from instantaneous slip has an earlier arrival time with larger amplitudes compared to the observations. However this difference becomes smaller for distant stations to the south similar to Fukushima and more insignificant for other more distant stations. Satake *et al.* (2013) also reported that the large slip in the vicinity of the trench axis is responsible for the large coastal tsunami heights along the Sanriku coast located between 38.5° N to 40° N latitude which is shown in Figure 2.5. The tsunamis estimated from the deep subfaults arrive earlier and generate a gradual sea level uplift, while the tsunami waveforms computed from the shallow slip show later arrivals and reproduce the large impulsive tsunamis as shown for TM1, TM2, and Iwate south. The shallow slip produces a large seafloor deformation with a short wavelength which cannot inundate far distances overland. But, the slip on the deep region of the fault can produce the seafloor deformation with a longer wavelength leading to a larger period which can penetrate inland.

Many studies which have modeled the 2011 Tohoku tsunami mainly consider the slip earthquake fault. The published results for tsunami propagation and impact on the coastal region underpredict the field measurement height of up to 40 m along the Sanriku district in Tohoku region (Figure 2.5). Also the waveforms of three GPS buoys near Iwate prefecture which show a high frequency in the observations cannot be simulated properly. Tappin *et al.* (2014) discuss the most likely additional tsunami source known as submarine mass failure which may address these gaps. The study of pre and post tsunami surveys based on vertical seafloor movement with slope instability confirms that the horizontal acceleration from the earthquake can potentially trigger a submarine landslide. This study showed the tsunami run-up and inundation along the Sanriku coast by combining the FEM source model with submarine mass failure (SMF) can reproduce the field measurements better, compared to the seismic source alone. The combined initialization known as dual source could reproduce the higher frequency waves of the offshore GPS buoys and the DART buoys more successfully compared to those published models based on only an earthquake source. The dispersive part of the waveforms normally need a dispersive tsunami model as the long wave equations cannot predict the oscillation feature of the waves both for the far field and near field stations which are also reported in (Saito *et al.*, 2011).

Grilli *et al.* (2012) simulated the 2011 Tohoku-Oki tsunami by coseismic tsunami sources based on inverting onshore and offshore geodetic data, using a 3D finite element model (FEM) instead of the Okada (1985) planar fault model. They used the fully nonlinear and dispersive Boussinesq wave model (FUNWAVE-TVD). FUNWAVE uses a hybrid numerical scheme by finite volume TVD method for the non-linear SWEs with a finite difference treatment for higher order dispersive terms. For the wave breaking case, the dispersive terms are turned off to switch to non-linear SWEs. It was concluded that for the near field region of the Japan test case the dispersive effects are negligible due to short propagation distances and coarse grid resolution but may become more important in the far-field of the deep ocean for 20-40% of tsunami amplitude. A further discussion and analysis of the dispersive effects and Coriolis force for the 2011 tsunami event can be found in (Kirby *et al.*, 2013). It should be added that Boussinesq models, because of their more complex equations, are typically more computationally demanding than non-linear SWEs. In this regard, the FUNWAVE model is parallelized by the Message Passing Interface (MPI) method via the horizontal decomposition of the domain; otherwise running such models over a large domain with sufficiently fine resolution is impractical. The simulation was also repeated by the standard seismic inversion sources known

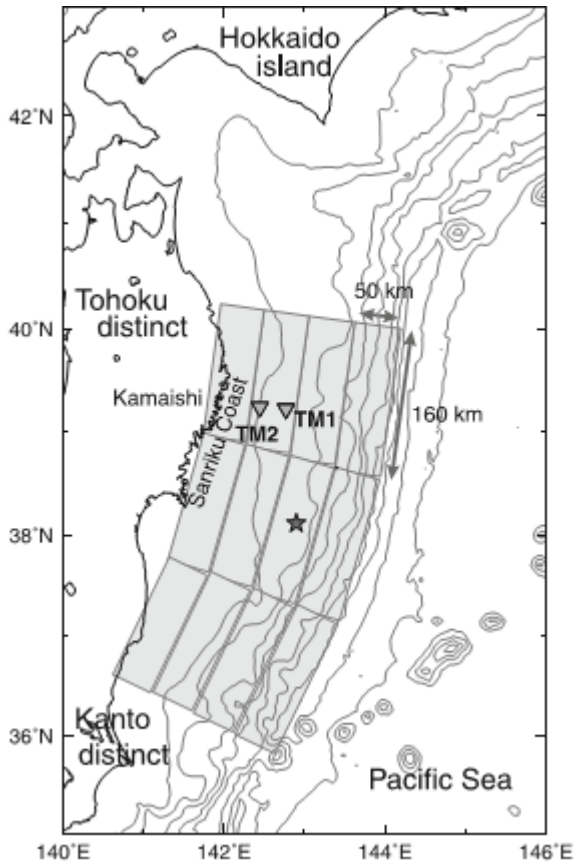


Figure 2.5 The Sanriku coast in Tohoku region, is located between 38.5° N and 40° N (Maeda *et al.*, 2011). Ocean bottom pressure gauges (OBP) called TM1 and TM2 off Kamaishi, Northern Japan, shown with triangles.

as University of California, Santa Barbara (UCSB) coseismic source with instantaneous and time-dependent methods. However, the results of the waveform for nine GPS buoys near Japan show that it is not easy to judge the superiority of these initial models. The USCB fault model based on Okada shows better agreement for three gauges, but for another three gauges the transient FEM model shows better agreement compared to the USCB fault model, and neither source is capable to reproduce the shorter wave oscillations related to the Iwate region which has a closer distance to the epicenter. Overall, good agreement was seen for both initial models, however, the USCB overpredicts the leading wave crest and the amplitude of the leading wave troughs. Both sources predict the arrival time of the lead wave slightly earlier than observations.

Popinet (2012) implemented the quadtree adaptive grid for the long distance wave propagation and flooding of the Tohoku tsunami. The hydrodynamic model solves the non-linear SWEs and the source model is chosen directly from the seismic inversions from (Shao *et al.*, 2011). Popinet (2012) selected the seismic source wave only, so that the comparisons between the predicted and observed waveforms are independent validation of both source and tsunami model. The vertical displacement is achieved by superposition of the 190 Okada subfault displacements, while the rupturing time is neglected. Different levels of refinement are applied from 250 km to 250 m and for 5 hour of simulation the number of grids reached one million. Popinet (2012) reported for the 2011 Japan tsunami simulation, the gain in runtime by the proposed quadtree implementation would be a factor from 3 to 10. Overall, the runtime required to predict the tsunami arrival time by the model is about two times greater than the physical time. Therefore the runtime gain by adaptive grid is not sufficient, and use of parallel techniques is recommended, including dynamic parallel load balancing (Agbaglah *et al.*, 2011). The model performed better agreement for most DART buoys compared to other gauges considering that DART records were not used to constrain the source model. It is also argued that the simulation of tsunami wave at tide gauges is more difficult compared to the open sea buoys, as these gauges are usually affected by the resonance of the harbours or small scale bays. Despite this phenomenon, it was shown that the arrival time and amplitude of the first wave could be predicted in most gauges. Sites similar to Omaezaki station are influenced by geometry of the harbour which can amplify the wave in reality and it is consequently underestimated by the models (Abe, 2011).The wave amplitudes at six GPS buoys and two bottom pressure gauges are also included in the results. The wave amplitude for the buoy Iwate central is overestimated, and Miyagi north and Miyagi central are underestimated by a factor of two. It was proposed that the discrepancies of these gauges are due to biases as a result of the source model rather than the hydrodynamic model. These differences also reported by other researchers in the literature.

For example, the source model proposed by Fujii *et al.* (2011), although is based on the inversion of waveforms, still underpredicts the wave amplitude in the Miyako region.

MacInnes *et al.* (2013) also carried out a study of the Tohoku 2011 earthquake to study how different tsunami sources based on seismic inversions, tsunami waveforms and GPS data can affect the near field predictions. The Geoclaw software based on a finite volume solver is selected to simulate the tsunami wave propagation and inundation. By comparing 10 sources, the simulation results showed agreement within 20% difference from observations including the large inundation in the Sendai plain, confirming the large initial ocean floor uplift near the trench. Most of the earthquake sources, even with high grid resolution, could not reproduce the high run-up between 39° to 40° N, suggesting that an extra source of tsunamigenic energy might have been involved (e.g., earthquake induced landslides, lateral motion of the coastline or bathymetric features) to explain the observations as also mentioned by Grilli *et al.* (2012). Different resolutions attempted to simulate the wave propagation, and the results did not show a significant difference of the wave amplitude such as damping or amplification, supporting the concept that fine bathymetry resolution is mainly important for the inundation process and predicting the wave height on land.

Wei *et al.* (2014) assessed the performance of three types of source models including tsunameter, GPS tsunameter and the USGS finite fault model for the 2011 Japan tsunami. The aim of this was to address the merits and limitations of these tsunami sources and investigate their potential for accurate and real time forecasting. Tsunameters are the earliest direct tsunami observations where the measurements are recorded from water pressure changes from the tsunami waves. These tsunameters are located offshore, and therefore are not affected by harbour or other short wavelength waves. Also pressure information from deep ocean stations follows long wave theory which allows for rapid tsunami source inversions (Wei *et al.*, 2003; Percival *et al.*, 2011). Many successful real-time forecasts since the 2004 Indian ocean tsunami have been reported e.g., (González *et al.*, 2005) and currently the deep ocean measurements are the main component of NOAA's tsunami forecast system. Such tsunameters are located near subduction zones 30-60 min tsunami travel time from the tsunamigenic earthquake sources to avoid the nonlinear and dispersion waves from the shallow water interface. On the other hand, the first 30 min is critical for the near-field tsunami forecasting, implying the limitations of these measurements. In this regard, after the December 2012 Japanese earthquake, some more tsunameters were deployed off the shore of Japan which provide the measurements within 10-20 min (Bernard *et al.*, 2014). This can improve and shorten the detection of the tsunami time significantly by locating the tsunameters closer to potential sources (Wei *et al.*, 2013).

The data from GPS can record measurements of ground displacements from a triggered earthquake which will provide the earthquake slip parameters shortly after large events (Blewitt *et al.*, 2006; Simons *et al.*, 2011). The limitation for GPS data is mainly related to their restriction over land areas apart from a few underwater observation points (Sato *et al.*, 2011). Therefore to only depend on the results from GPS may not be sufficient to characterize the earthquake slip which occurs offshore and near the trench. Thus the measurements from deep ocean tsunameters are complementary to GPS data for a better estimation of the tsunami generation process (Wei *et al.*, 2014). In the near future the combination of various source measurements including GPS with seismic and tsunami data can potentially deliver the best rapid prediction for earthquake rupture and tsunami generating process.

The investigation of tsunami hazard in the literature has been more focused to the tsunami characteristics such as tsunami wave height, arrival time, run-up and inundation. Recently, the measurement of tsunami current velocity has also become to the attention for the assessment of the tsunami hazards for both near-field and far-field tsunamis. For example, the observations show even without flooding, the maritime assets are vulnerable to severe damage from powerful currents induced by tsunami surges, which is less studied or understood (Wilson *et al.*, 2013; Lynett *et al.*, 2014). Lynett *et al.* (2014) illustrated in their study that the current tsunami velocities between 1.5 to 3 m/s can cause moderate damage, velocities more than 3 m/s can cause major damage and velocities greater than 4.5 m/s can cause extreme damage to the ports and harbors. Fritz *et al.* (2012) measured the velocity of the 2011 Japan tsunami up to 11 m/s at Kesennuma Bay narrows in Miyagi prefecture which proves the sever impact of this tsunami in the Tohoku region. Lynett *et al.* (2014) studied the impact of the 2011 Tohoku tsunami to the Crescent city in California using the MOST (shallow flow model) and COULWAVE (Boussineq model). It was shown that both models can predict the velocities well compared to the tide station data. It was stated, although the higher-order COULWAVE model could estimate more accurate velocities, MOST model is preferred as it is less computationally demanding and the results found to be conservative in terms of hazard assessment.

Arcos and LeVeque (2015) validated the velocities of the GeoClaw model by using the observations near Hawaii from the 2011 Japan tsunami. The authors, stated that the recorded observations show that the velocity is not uniform along the water depth in many locations. Therefore, the tsunami models based on SWEs similar to GeoClaw software, might not be suitable for modeling such flows. Nevertheless, the velocity results from the GeoClaw model could successfully match the depth-averaged velocities of the observations at most gauges. Although, there are some locations around the harbor which could not be captured by the model.

The authors argued this could be due to the lack of sufficient accurate bathymetry and the fact that the flows might be more complex, 3D, and turbulent e.g. inside the harbors which cannot be predicted accurately by SWEs. The important point of this study is the sensitivity of the velocities to the spatial location. It is demonstrated that, tsunami current velocity is more spatially variable than waveform or height and, therefore, may be a more sensitive variable for model validation. It should be noted that similar studies and results regarding the effect of 2011 Japan tsunami currents across the Pacific Ocean around the Hawaiian Islands and along the Honolulu coast have been reported by Cheung *et al.* (2013) and Yamazaki *et al.* (2012) respectively, who used the NEOWAVE model for their simulations. In the aforementioned studies, in order to resolve inside the harbors, reefs, channels and shores around the gauges, fine grid resolution going down to 10 m was used. The necessity of using fine resolution grids implies high computational costs. Therefore, a computationally efficient tsunami model is a privilege for prediction of the tsunami current velocities.

2.6 Summary

This chapter has reviewed different hydrodynamic tsunami models and numerical schemes to solve those hydrodynamic models. The main hydrodynamic models which have been used for tsunami simulations are including SWE and Boussinesq models. It was discussed that Boussinesq type models are more important for the wave propagation in far field tsunamis. Otherwise SWE models perform fairly well for earthquake generated tsunamis comparable to non-hydrostatic or Boussinesq models. On the other hand, it was discussed the Boussinesq models are computationally much more demanding, due to the implicit nature of the solution technique used to deal with dispersion terms compared to SWEs. Hence, SWE models are preferred for real time tsunami simulations. Various numerical schemes including finite difference, finite volume, finite element, and SPH have been reported to be used to solve the hydrodynamic models. Each numerical scheme has its own advantages and limitations. Concerning the computational time, as summarized in the literature, finite difference and finite volume are the best options compared to other numerical schemes. However, finite volume method solutions are becoming more popular as they take advantage of the conservation property, and auto-shock capturing capability which can handle different types of flows. This could be a useful scheme to solve the non-linear SWE both for the propagation and inundation as the high resolution finite volume Godunov type schemes are very robust. In this regard, in this research the second order MUSCL-Hancock scheme is selected including the techniques to tackle with well-balanced problem and wetting/drying over irregular topography.

Computational time is crucial for real tsunami prediction. SWE models, however, require less computational effort compared to other alternative models, but, due to their explicit nature, they are constrained with small time step to satisfy the numerical stability which may lead to longer computational time than the real tsunami travel time. Hence, different parallel computing techniques reviewed in the context of tsunami simulations. It was reported that shared memory parallelization using modern GPU technology could save the simulation time significantly in different scientific applications. In this regard, a single GPU is potentially capable to save the computational time by order of magnitudes compared to a single CPU. Therefore, it seems GPU parallel computing technology is cheaper compared to conventional CPU parallelization on clusters, offering a similar performance. However, fewer implementations have been reported in the context of tsunami modelling. Therefore, the GPU parallel computing technique is implemented in the proposed hydrodynamic model.

The adaptive grid techniques which have been used for multi-scale tsunami simulations are also reviewed. Nesting grid approach is mainly used for the conventional tsunami models which solve the SWE by finite difference scheme e.g. TUNAMI, MOST, and COMCOT. Adaptive mesh refinement (AMR) including block adaptive refinement (e.g., GeoClaw software) and hierarchical methods are employed with the finite volume schemes. It was discussed the AMR technique require computational overheads to search and save the data structures. Therefore, a simplified AMR method with no data structure is recommended to be used for the CPU based model in this work. Implementation of adaptive grids on GPU with a finite volume solver in the context of tsunami modelling has not been yet reported in the literature. In this research, a static adaptive mesh refinement will be implemented to the finite volume solver on GPU which will be presented in Chapter 6.

The source estimation techniques of tsunamis which are generated by earthquake has been briefly described. It was mentioned that the real time GPS has the ability to provide tsunami early warning in few minutes. The research on GPS technology is under way, however, it is recommended to use the GPS data in combination by other inversion-based slip approaches for a more accurate source estimation. These approaches are including, teleseismic data, tsunami waveform data and tsunameter data from deep Ocean. In this research the two latter approaches selected to reproduce the 2011 Japan tsunami. Finally some of the recent researches on 2011 Japan tsunami using different tsunami models and numerical schemes are also reviewed.

Next chapter introduces a hydrodynamic tsunami model which is capable to simulate the tsunami from propagation to inundation. The finite volume Godunov type scheme and further numerical techniques used are explained. Then two different high performance computing techniques using CPU and GPU are implemented. The model will be validated in Chapter 4.

Chapter 3 Finite Volume Godunov Type Tsunami Model

This chapter describes the hydrodynamic model that is adopted and further developed in this study for tsunami modelling. In recent years, shallow flow simulations have been more focused on developing models for real world problems. The aim of this research is to develop a computationally efficient shallow flow model for real world tsunami simulations. In this regard the finite volume Godunov type scheme is adopted to solve the shallow water equations (SWEs) to provide automatic shock-capturing capability and describe the complex tsunami induced shallow wave hydrodynamics. The pre-balanced SWEs which automatically maintains the well-balanced solutions for wet-bed application are solved by the aforementioned finite volume Godunov-type scheme incorporated with a Harten, Lax, and van Leer with the contact-restoration (HLLC) approximate Riemann solver for interface flux calculation. The higher order accuracy in time and space is achieved by using the two-step unsplit MUSCL-Hancock method.

The numerical technique used to ensure well-balanced non-negative solution for wetting and drying over complex bathymetry/topographies is reviewed and explained in Section 3.1. Then the two high-performance computing techniques implemented to accelerate the model are introduced in Section 3.3 and 3.4. Model validation and performance assessment will be presented and discussed in detail in Chapter 4.

3.1 Governing equations and numerical scheme

As discussed in Chapter 2 the shallow water equations (SWEs) are commonly accepted and widely used for tsunami modelling and other engineering applications. To support these applications, it requires a SWE model to be able to deal with complex flow hydrodynamics such as capturing shocks and bores and representing wetting and drying (moving shoreline) over complex domain topography with a well-balanced solution. These challenges are resolved by adopting a robust finite volume shock-capturing numerical model which will be reviewed herein.

Godunov-type methods entail an essential property of hyperbolic partial differential equations, namely wave propagation information, to construct the numerical schemes. The resulting Godunov-type numerical models provide automatic shock-capturing capability (LeVeque, 1992). Finite volume method incorporated with a Godunov-type scheme provides a robust means to solve the SWEs for complex shallow flow hydrodynamics. Numerous finite volume

Godunov-type schemes have been developed and reported for shallow flow simulations in the last three decades.

The shallow water equations (SWEs) may be derived by integrating in depth the Reynolds-averaged Navier-Stokes equations, assuming long wave propagation and negligible vertical water particle acceleration. Toro (2001) reviewed the different types of the full SWEs, which may be commonly written in a matrix form of hyperbolic conservation laws as:

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = \mathbf{S}(\mathbf{U}) \tag{3.1}$$

where x, y and t respectively represent the two Cartesian coordinates and time, $\mathbf{U}$ is the vector of conserved variables, $\mathbf{F}(\mathbf{U})$ and $\mathbf{G}(\mathbf{U})$ are the flux vectors and $\mathbf{S}(\mathbf{U})$ contains the source terms. Viscous terms, surface stresses are ignored for tsunami simulations. Coriolis force could be neglected as long as the tsunami travel distance remains insignificant compared to the earth's complete rotation time; otherwise over long distances the compounding effect of Coriolis force may sum up and increase (Kowalik *et al.*, 2005; Eze *et al.*, 2009). By ignoring these terms, the vectors of the pre-balanced formulation of SWEs are given as follows (Liang and Borthwick, 2009):

$$\mathbf{U} = \begin{bmatrix} \eta \\ uh \\ vh \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} uh \\ u^2 h + 0.5g(\eta^2 - 2\eta z_b) \\ uvh \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} vh \\ uvh \\ v^2 h + 0.5g(\eta^2 - 2\eta z_b) \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 \\ -c_f u\sqrt{u^2+v^2} - g\eta \dfrac{\partial z_b}{\partial x} \\ -c_f v\sqrt{u^2+v^2} - g\eta \dfrac{\partial z_b}{\partial y} \end{bmatrix} \tag{3.2}$$

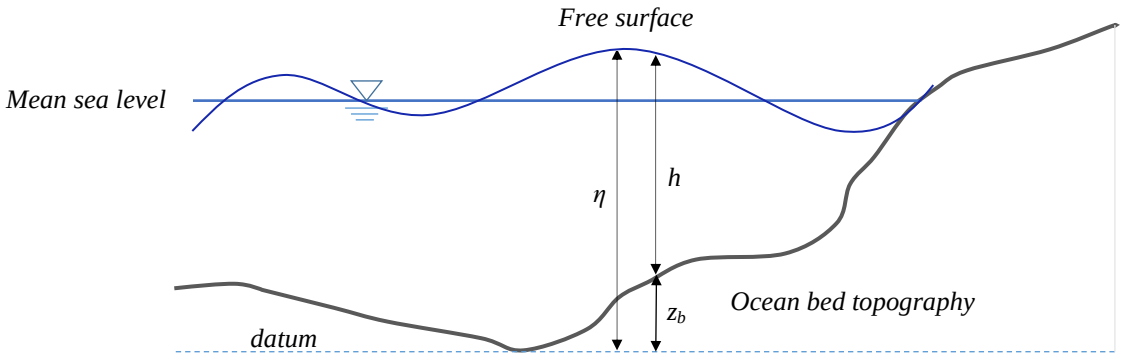


Figure 3.1 Sketch of the bathymetry/topography and free surface elevation.

where η is the water level (stage), h is the total water depth, and z_b is the bed level above datum, as shown by the definition sketch in Figure 3.1; u and v are depth-averaged velocities in the x- and y-direction; uh (=qx) and vh (=qy) are the unit-width discharge in the x- and y-directions

respectively. $C_f = gn^2 / h^{(1/3)}$ is the bed roughness coefficient with n denoting the Manning coefficient; and $g = 9.81\ m/s^2$ the acceleration due to gravity. Using the water level η as a flow variable, the above formulation provides a set of pre-balanced SWEs that are able to automatically preserve the lake at rest solution at the discrete level for wet-bed applications involving irregular domain topographies.

The pre-balanced shallow water equations (3.1)-(3.2) may be solved using a first-order accurate explicit finite volume Godunov-type approach that updates the flow variables to a new time step using the following formula:

$$\mathbf{U}_{i,j}^{k+1} = \mathbf{U}_{i,j}^{k} - \frac{\Delta t}{\Delta x_{i,j}}\left(\mathbf{F}_{i+1/2,j} - \mathbf{F}_{i-1/2,j}\right) - \frac{\Delta t}{\Delta y_{i,j}}\left(\mathbf{G}_{i,j+1/2} - \mathbf{G}_{i,j-1/2}\right) + \Delta t\,\mathbf{S}_{i,j} \tag{3.3}$$

where subscript i and j are the cell indices; superscript k represents the time level; Δt, $\Delta x_{i,j}$ and $\Delta y_{i,j}$ are the time step and cell sizes in the x and y-directions, respectively; $\mathbf{F}_{i+1/2}$, $\mathbf{F}_{i-1/2}$, $\mathbf{G}_{j+1/2}$ and $\mathbf{G}_{j-1/2}$ are the interface fluxes through the east, west, north and south edges of each computational cell, which are evaluated by solving local Riemann problems using an HLLC approximate Riemann solver (Toro *et al.*, 1994).

In a first-order scheme, the face values of flow variables are considered to be directly the same as the values of the cell-centred flow variables. At the left hand side of the east cell interface (i +1/2, j), the face values are given by:

$$\begin{aligned}
&\bar{\eta}^{L}_{i+1/2,j} = \eta_{i,j}\,,\ \bar{h}^{L}_{i+1/2,j} = h_{i,j} \\
&\bar{q}^{L}_{x\,i+1/2,j} = q_{x\,i,j}\,,\ \bar{q}^{L}_{y\,i+1/2,j} = q_{y\,i,j} \\
&\bar{z}^{L}_{b\,i+1/2,j} = \bar{\eta}^{L}_{i+1/2,j} - \bar{h}^{L}_{i+1/2,j}
\end{aligned} \tag{3.4}$$

Similarly, the face values at the right hand side of the east cell interface (i +1/2, j) are given by:

$$\begin{aligned}
&\bar{\eta}^{R}_{i+1/2,j} = \eta_{i+1,j}\,,\ \bar{h}^{R}_{i+1/2,j} = h_{i+1,j} \\
&\bar{q}^{R}_{x\,i+1/2,j} = q_{x\,i+1,j}\,,\ \bar{q}^{R}_{y\,i+1/2,j} = q_{y\,i+1,j} \\
&\bar{z}^{R}_{b\,i+1/2,j} = \bar{\eta}^{R}_{i+1/2,j} - \bar{h}^{R}_{i+1/2,j}
\end{aligned} \tag{3.5}$$

The corresponding x- and y-direction velocity components are then evaluated by:

$$\begin{aligned}
&\bar{u}^{L}_{i+1/2,j} = \bar{q}^{L}_{x\,i+1/2,j} \Big/ \bar{h}^{L}_{i+1/2,j}\,,\ \bar{u}^{R}_{i+1/2,j} = \bar{q}^{R}_{x\,i+1/2,j} \Big/ \bar{h}^{R}_{i+1/2,j} \\
&\bar{v}^{L}_{i+1/2,j} = \bar{q}^{L}_{y\,i+1/2,j} \Big/ \bar{h}^{L}_{i+1/2,j}\,,\ \bar{v}^{R}_{i+1/2,j} = \bar{q}^{R}_{y\,i+1/2,j} \Big/ \bar{h}^{R}_{i+1/2,j}
\end{aligned} \tag{3.6}$$

If the water depth is less than 10^{-6} m, the cell under consideration is defined as dry and the velocities in (3.6) all become zero.

As an important step at each cell interface, a single bed elevation is defined on the basis of above face values for obtaining the Riemann states (Audusse *et al.*, 2004). Taking cell interface (i +1/2, j) as an example

$$z_{b\,i+1/2,j} = \mathbf{max}\left(\bar{z}^{L}_{b\,i+1/2,j}, \bar{z}^{R}_{b\,i+1/2,j}\right) \tag{3.7}$$

The Riemann states of the water depth can be subsequently reconstructed as:

$$h^{L}_{i+1/2,j} = \mathbf{max}\left(0, \bar{\eta}^{L}_{i+1/2,j} - z_{b\,i+1/2,j}\right), \quad h^{R}_{i+1/2,j} = \mathbf{max}\left(0, \bar{\eta}^{R}_{i+1/2,j} - z_{b\,i+1/2,j}\right) \tag{3.8}$$

which obviously ensures non-negativity of water depth. The other Riemann states can be evaluated accordingly:

$$\begin{aligned}
&\eta^{L}_{i+1/2,j} = h^{L}_{i+1/2,j} + z_{b\,i+1/2,j}\,,\ \eta^{R}_{i+1/2,j} = h^{R}_{i+1/2,j} + z_{b\,i+1/2,j} \\
&q^{L}_{x\,i+1/2,j} = \bar{u}^{L}_{i+1/2,j} h^{L}_{i+1/2,j}\,,\ q^{R}_{x\,i+1/2,j} = \bar{u}^{R}_{i+1/2,j} h^{R}_{i+1/2,j} \\
&q^{L}_{y\,i+1/2,j} = \bar{v}^{L}_{i+1/2,j} h^{L}_{i+1/2,j}\,,\ q^{R}_{y\,i+1/2,j} = \bar{v}^{R}_{i+1/2,j} h^{R}_{i+1/2,j}
\end{aligned} \tag{3.9}$$

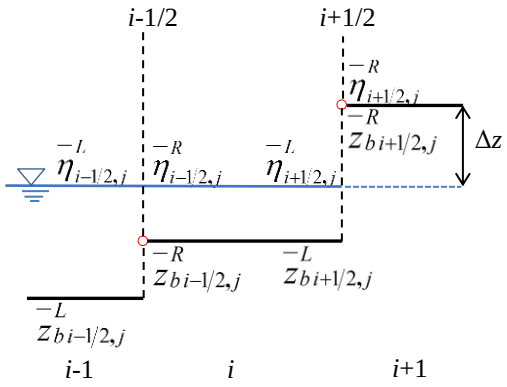


Figure 3.2 Well-balanced scheme for dry-bed applications.

Combined with the pre-balanced governing equations, the Riemann states reconstructed following the above procedure can guarantee non-negative water depth and maintain lake at rest solution except for one case as shown in Figure 3.2. In Figure 3.2, a wet cell (i, j) shares a common interface (i+1/2, j) with a dry cell at (i+1, j) at the discrete level; the bed level at (i+1, j) is higher than the water surface level at (i, j); this case must be treated with a proper numerical technique to satisfy the well-balanced property, as proposed by Liang (2010).

Following the non-negative reconstruction at (i+1/2, j), the bed profile, water depth and free surface elevation would be given as follows

$$
\begin{aligned}
&z_{b\,i+1/2,j} = \bar{z}^{R}_{b\,i+1/2,j} \\
&h^{L}_{i+1/2,j} = h^{R}_{i+1/2,j} = 0 \\
&\eta^{L}_{i+1/2,j} = \eta^{R}_{i+1/2,j} = z_{b\,i+1/2,j}
\end{aligned}
\tag{3.10}
$$

The fluxes across (i+1/2, j) are evaluated based on the ground level rather than the actual still water level at cell (i, j) which is a constant value (η) for a lake at rest problem. However, the fluxes in (i-1/2, j) are approximated according to constant water level η. This subsequently leads to spurious net fluxes and drives the flow into motion at cell (i, j) at the next time step, violating the well-balanced property. To overcome this problem, Liang (2010) proposed a local bed modification approach by first finding the difference Δz between the actual water level at wet cell and the calculated water level (i.e. the ground level at the dry cell):

$$
\Delta z = \max\left(0, \quad (z_{b\,i+1/2,j} - \bar{\eta}^{L}_{i+1/2,j})\right) \tag{3.11}
$$

This difference is then subtracted from the original values of the bed elevation and the stage components of Riemann states as follows:

$$
\begin{aligned}
&z_{b\,i+1/2,j} - \Delta z \longrightarrow z_{b\,i+1/2,j} \\
&\eta^{L}_{i+1/2,j} - \Delta z \longrightarrow \eta^{L}_{i+1/2,j} \\
&\eta^{R}_{i+1/2,j} - \Delta z \longrightarrow \eta^{R}_{i+1/2,j}
\end{aligned}
\tag{3.12}
$$

It is clear after the local modification the well-balanced property is satisfied as

$$
\eta^{L}_{i+1/2,j} = \eta^{R}_{i+1/2,j} = z_{b\,i+1/2,j} = \eta \tag{3.13}
$$

The bed modification in (3.12) is the final stage of the nonnegative reconstruction of Riemann states. The modified Riemann states are then employed by the HLLC approximate Riemann solver for the calculation of the interface fluxes.

3.1.1 *HLLC approximate Riemann solver*

The HLLC approximate Riemann solver (Toro *et al.*, 1994) is implemented in this work because it is easy to implement for applications involving wetting and drying and it can automatically satisfy the entropy condition. More details about approximate Riemann solvers are given by (Toro, 2001; Toro, 2009). It should be noted that approximate Riemann solvers require less

computational effort than the exact solvers but can achieve a proper degree of accuracy for most shallow flow simulations (Erduran *et al.*, 2002; Zoppou and Roberts, 2003).

The solution structure of HLLC approximate Riemann solver is shown in Figure 3.3. Taking the east interface (i+1/2, j) of cell (i, j) as an example, the interface fluxes $\mathbf{f}_{i+1/2,j}$, are evaluated by:

$$\mathbf{f}_{i+1/2,j} = \begin{cases} \mathbf{f}_L & if \quad 0 \le S_L \\ \mathbf{f}_{*L} & if \quad S_L \le 0 \le S_M \\ \mathbf{f}_{*R} & if \quad S_M \le 0 \le S_R \\ \mathbf{f}_R & if \quad 0 \ge S_R \end{cases} \tag{3.14}$$

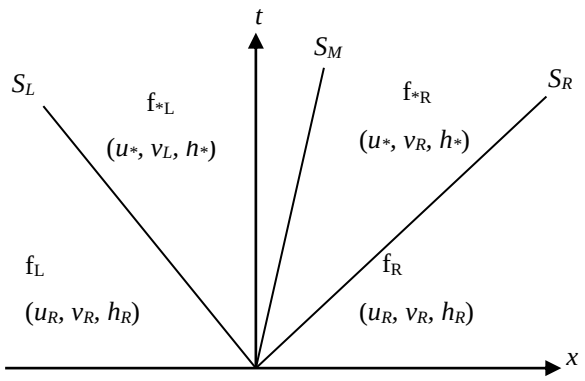


Figure 3.3 The solution structure of the HLLC approximate Riemann solver.

According to the solution region division in Figure 3.3, the fluxes in the left and right regions are (f_L, f_R) while the left and right fluxes in the middle region (known also as star region) are shown as (f_{*L}, f_{*R}) and evaluated as:

$$\mathbf{f}_{*L} = \begin{bmatrix} \mathbf{f}_{1*} \\ \mathbf{f}_{2*} \\ \mathbf{f}_{1*} \cdot v_L \end{bmatrix} \quad \text{and} \quad \mathbf{f}_{*R} = \begin{bmatrix} \mathbf{f}_{1*} \\ \mathbf{f}_{2*} \\ \mathbf{f}_{1*} \cdot v_R \end{bmatrix} \tag{3.15}$$

The first and second entries in the above middle region flux vectors are evaluated by the HLL (Harten, Lax, van Leer) formula as proposed by Harten *et al.* (1983).

$$\mathbf{f}_* = \frac{S_R \mathbf{f}_L - S_L \mathbf{f}_R + S_L S_R (\mathbf{u}_R - \mathbf{u}_L)}{S_R - S_L} \tag{3.16}$$

In equation (3.14), S_L and S_R are the smallest and largest wave speeds in the solution structure while S_M is the middle wave speed. Different options for wave speed estimates are available. Herein the wave speed approximations involving a two-rarefaction approximate Riemann solver including the dry-bed case suggested by Fraccarollo and Toro (1995) are used:

$$S_L = \begin{cases} u_R - 2\sqrt{gh_R} & if \quad h_L = 0 \\ \min\left(u_L - \sqrt{gh_L}, u_* - \sqrt{gh_*}\right) & if \quad h_L > 0 \end{cases} \tag{3.17}$$

$$S_R = \begin{cases} u_L + 2\sqrt{gh_L} & if \quad h_R = 0 \\ \max\left(u_R + \sqrt{gh_R}, u_* + \sqrt{gh_*}\right) & if \quad h_R > 0 \end{cases} \tag{3.18}$$

where h_L, h_R, u_L and u_R are the components of the left and right Riemann states for a local Riemann problem, and u_* and h_* are evaluated from:

$$u_* = \frac{1}{2}(u_L + u_R) + \sqrt{gh_L} - \sqrt{gh_R} \tag{3.19}$$

$$h_* = \frac{1}{g}\left[\frac{1}{2}\left(\sqrt{gh_L} + \sqrt{gh_R}\right) + \frac{1}{4}(u_L - u_R)\right]^2 \tag{3.20}$$

For a general problem including wetting and drying, Toro (2001) recommended computing the middle wave speed using the following formula:

$$S_M = \frac{S_L h_R (u_R - S_R) - S_R h_L (u_L - S_L)}{h_R (u_R - S_R) - h_L (u_L - S_L)} \tag{3.21}$$

In a similar way the Riemann states and corresponding fluxes on the other cell interfaces can be approximated.

3.1.2 *Discretisation of source terms*

To update the flow variables to the next time-step, the source term must also be evaluated in a compatible way to maintain the well-balanced solution and non-negative water depth. The key components in the source term vector of **S**(**U**) include the bed slope and friction source terms. Due to the use of the pre-balanced governing equations (3.1)-(3.2), the bed slope source terms can be simply discretised and solved by a central difference scheme. For example, the *x*-direction slope source term may be discretised as:

$$-g\eta\frac{\partial z_b}{\partial x} = -g\bar{\eta}\left(\frac{z_{bi+1/2,j} - z_{bi-1/2,j}}{\Delta x}\right) \quad \text{with} \quad \bar{\eta} = \left(\eta^R_{i-1/2,j} + \eta^L_{i+1/2,j}\right)/2 \tag{3.22}$$

It should be noted that the stage and bed elevation values are those obtained after imposing the local bed modification in (3.12). The *y*-direction slope source term can be discretised in a similar way.

As a whole, the above finite volume Godunov-type scheme is able to provide well-balanced non-negative solutions for different types of shallow flow hydrodynamics over complex topography. In addition to the local bed modification technique, the overall numerical scheme does not need any 'numerical tricks' to ensure non-negative depth for applications involving wetting and drying, unlike those used by Brufau *et al.* (2004) and Liang and Borthwick (2009) which must modify the flow variables when a wet-dry interface is present.

For tsunami simulations, bed friction can be an important factor for the flow evolution in the inundation zone and for run-up estimations, although it is less important for the wave propagation in the deep ocean. To avoid the probable numerical instability caused by the inclusion of the friction source terms, the splitting point-implicit scheme (Bussing and Murman, 1988; Fiedler and Ramirez, 2000) is implemented in the current model, which involves solving the following ordinary differential equation (ODE):

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} = \mathbf{S}_f \tag{3.23}$$

where $\mathbf{S}_f = \begin{bmatrix} 0 & S_{fx} & S_{fy} \end{bmatrix}^{\mathrm{T}}$ with $S_{fx} = -c_f u\sqrt{u^2+v^2}$ and $S_{fy} = -c_f v\sqrt{u^2+v^2}$. In equation (3.23), the friction has no effect on the continuity equation and therefore only the solutions for the momentum components are attempted.

Equation (3.23) in the *x*-direction can be discretized by the proposed implicit method as:

$$\frac{q_x^{k+1} - q_x^k}{\Delta t} = S_{fx}^{k+1} \tag{3.24}$$

by expanding S_{fx}^{k+1} with a Taylor series:

$$S_{fx}^{k+1} = S_{fx}^k + \left(\frac{\partial S_{fx}}{\partial q_x}\right)^k \Delta q_x + o(\Delta q_x^2) \tag{3.25}$$

where $\Delta q_x = q_x^{k+1} - q_x^k$. By ignoring higher order terms and substituting it into Equation (3.24), the following time marching formula for q_x is obtained:

$$q_x^{k+1} = q_x^k + \Delta t\left(\frac{S_{fx}}{D_x}\right)^k = q_x^k + \Delta t F_x \tag{3.26}$$

where $D_x = 1 - \Delta t\left(\partial S_{fx}/\partial q_x\right)^k$ is the implicit coefficient and F_x is the bed friction source term including the implicit coefficient D_x.

The inclusion of the friction term as stated before may affect the numerical stability near the wet dry front when the water depth disappears. To tackle this problem, a limiting value for F_x is derived by Liang and Marche (2009) as following:

$$F_x = \begin{cases} \geq -q_x^k / \Delta t & \text{if } q_x^k \geq 0 \\ \leq -q_x^k / \Delta t & \text{if } q_x^k \leq 0 \end{cases} \tag{3.27}$$

By imposing the above limiting value, the maximum effect of friction force is to stop the flow ($q^{k+1}q^k \geq 0$); if the value of F_x is evaluated beyond the permissible limit this value is replaced by the above critical value.

A similar method can be applied to get the friction in the *y*-direction. During the simulation process, at each time step *q* is updated by (3.26) to compute the friction effect for wet cells. The resulting flow variables are then used as initial conditions for the next time step, to be updated by the well-balanced numerical scheme discussed previously. The details of the overall implementation of the scheme can be found in (Liang, 2010; Wang *et al.*, 2011).

3.1.3 *Second order accuracy in space and time*

To construct a second-order accurate Godunov-type scheme in space and time for more accurate simulations, the unsplit MUSCL-Hancock method (van Leer, 1985) is applied, which involves a predictor step and a corrector step. The predictor step calculates intermediate flow variables over half of a time step using the following formula:

$$\mathbf{u}_{i,j}^{k+1/2} = \mathbf{u}_{i,j}^{k} - \frac{\Delta t}{2\Delta x}(\mathbf{f}_E - \mathbf{f}_W) - \frac{\Delta t}{2\Delta y}(\mathbf{g}_N - \mathbf{g}_S) + \frac{\Delta t}{2}\mathbf{s}_{i,j} \tag{3.28}$$

where the subscripts *E*, *W*, *N* and *S* indicate the east, west, north and south interfaces of the cell under consideration; *i*,*j* are the cell indices.

The flux terms $\mathbf{f}_{E,}$ $\mathbf{f}_w$, $\mathbf{g}_s$ and $\mathbf{g}_N$ are calculated from the face values at mid-points of cell faces, which are linearly reconstructed from the corresponding cell-centre values of flow variables using the MUSCL scheme:

$$\mathbf{u}(x,y) = \mathbf{u}_{i,j} + \psi(r)\mathbf{r}\nabla\mathbf{u}_{i,j} \tag{3.29}$$

where (*x*, *y*) is the midpoint of the cell face; **r** is the distance vector from the cell centre to point (*x*, *y*) with east and north being the positive in *x*- and *y*-direction, respectively; and $\nabla\mathbf{u}$ is the gradient vector. The predictor step does not need Riemann solutions and the fluxes are calculated within each cell by the reconstructed values at the extremities of the inner interfaces. A slope limiter is implemented to prevent spurious oscillations of the solution in the vicinity of

discontinuities or steep gradients. $\psi(r)$ slope limiter is a function of the ratio of successive gradients for the conserved variables. For example in the *x*-direction this ratio is defined as:

$$r = \Delta u_{upw} / \Delta u_{loc}$$
$$\Delta u_{upw} = u_E - u_{i,j}\ ,\quad \Delta u_{loc} = u_{i,j} - u_W \tag{3.30}$$

There are different choices for a slope limiter (Hirsch, 1995) which can be written as:

$$\psi(r) = \max\left[0, \min(\beta r, 1), \min(r, \beta)\right] \quad 1 \le \beta \le 2 \tag{3.31}$$

where β is the limiter parameter, $\beta = 1$ gives the minmod limiter, $\beta = 1.5$ gives the Sweby limiter and $\beta = 2$ representing the Roe's superbee limiter (Sweby, 1984; Roe, 1986). Minmod limiter is used throughout the simulations unless it is specifically mentioned. Generally, the minmod limiter is known to be more dissipative and gives smoother results than a superbee limiter.

To obtain better results, as reported by Hu *et al.* (2000), in the corrector step, the original flow variable values (u^k) instead of intermediate flow variables ($u^{k+1/2}$) are used for gradient vector terms and limiter in order to construct the cell interface for the Riemann states. In the corrector step, the HLLC approximate Riemann solver is used to calculate the interface fluxes and the flow variables are updated to a new time step using the following fully conservative time-marching formula (Toro, 2009):

$$\mathbf{u}_i^{k+1} = \mathbf{u}_i^k - \frac{\Delta t}{\Delta x}(\mathbf{f}_E - \mathbf{f}_W) - \frac{\Delta t}{\Delta y}(\mathbf{g}_N - \mathbf{g}_S) + \Delta t \mathbf{s}_i^{k+1/2} \tag{3.32}$$

Detailed implementation of this second-order finite volume HLLC Godunov-type scheme can be found in Liang and Borthwick (2009).

3.1.4 *Boundary conditions*

Two types of boundary conditions, i.e. transmissive (open) and closed (reflective), are used in this work.

Open boundary:

$$h_B = h_I,\quad u_B = u_I,\quad v_B = v_I \tag{3.33}$$

Closed boundary:

$$h_B = h_I,\quad u_B = -u_I,\quad v_B = v_I \tag{3.34}$$

where h is the water depth, u and v are the normal and tangential depth averaged velocity components. Subscripts B and I represent the positions at the boundary (ghost) and the inner cell adjacent to the boundary, respectively. This treatment of the boundary maintains the second-order accuracy (Hu *et al.*, 2000). For the open boundary conditions, the flow variables

from the inner boundary cell are copied to the ghost cell to allow zero gradients at the boundary; they might be also prescribed to impose inflow and outflow conditions. For the closed boundary conditions, the values for the normal velocity and the gradient of the water depth are computed to be zero at the boundary. To visualize the boundary condition, a computational domain is discretized with 'M' number of cells in one direction, including the ghost cells and inner cells as illustrated in Figure 3.4.

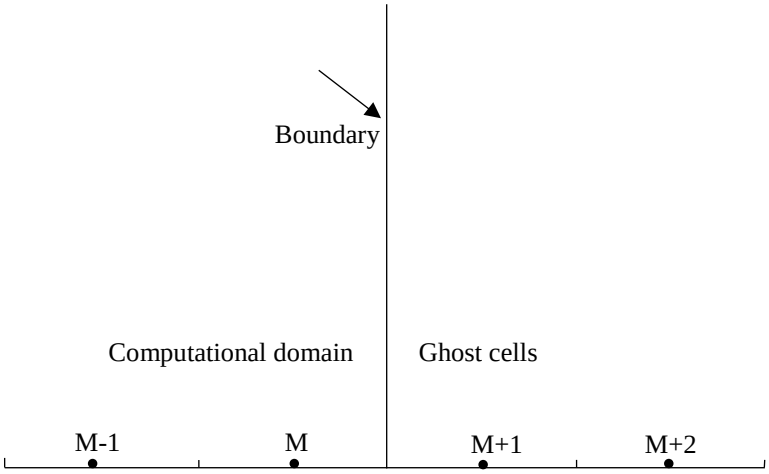


Figure 3.4 Computational cells inside and outside the boundary.

In Figure 3.4, 'M+1' and 'M+2' are the ghost cells outside the computational domain; the boundary values of *h* and *u* for a closed boundary are defined as below:

$$h_{M+1} = h_M \ , u_{M+1} = -u_M$$
$$h_{M+2} = h_{M-1} , u_{M+2} = -u_{M-1}$$

When the open boundaries involving subcritical flows present over irregular domain topographies, spurious waves are likely to occur adjacent to the boundary with a bump due to the missing information. This may consequently affect the flow solution inside the domain and cause numerical instability. To avoid this, the numerical approach proposed by Hou *et al.* (2015) is implemented in this work. In this approach the computational domain in the normal direction is extended by a few number of cells. The number of the required cells are dependent on the wave scale, geometry of the boundary and grid resolution. This boundary condition is used for the field scale and real world tsunami simulations in this project which will be presented in Chapter 4 and Chapter 5. More details of the aforementioned boundary treatment technique can be found in Hou *et al.* (2015).

3.1.5 *Stability criterion*

The current numerical scheme is overall explicit, and the maximum permissible timestep for stability is controlled by Courant-Friedrichs-Lewy (CFL) criterion (Courant *et al.*, 1967). The time step, Δt, is given as following:

$$\Delta t = C\min\left[\frac{\Delta x_{i,j}}{|u_{i,j}| + \sqrt{gh_{i,j}}}, \frac{\Delta y_{i,j}}{|v_{i,j}| + \sqrt{gh_{i,j}}}\right] \tag{3.35}$$

where $\Delta x_{i,j}$ and $\Delta y_{i,j}$ are the cell sizes in x- and y- directions; while $u_{i,j}$ and $v_{i,j}$ are the depth-averaged velocity components at cell (i, j); and C is the Courant number ($0 < C \leq 1$), which is set to 0.5 for all of the simulations in this work.

3.2 Parallel computing

In a simple term, parallel computing is to ensure many calculations are executed simultaneously. Parallelism has been developed for many years to support high-performance computing and the interest to it is growing due to the physical constraints of the computational power for those demanding numerical simulation tasks.

Traditional software are written for serial computation which a problem is solved by implementing a serial stream of instructions. By this concept one instruction may execute at a time and when that instruction completed, the next one is executed.

In contrast, parallel computing utilizes several processing elements simultaneously to solve a problem. In this regard, the problem should be divided into independent parts if possible to allow each processing element to execute its part of the algorithm simultaneously with others. Different processing elements exist, such as a single computer with multiple processors, computer clusters or combination of these. This defines the parallel and distributed computing.

In distributed computing, each processor has its own private memory and the information is exchanged between processors by passing messages.

In parallel computing, on the other hand, all processors can have access to a shared memory to exchange information between processors. Figure 3.5 shows a schematic view of a typical parallel system which each processor has access to a shared memory.

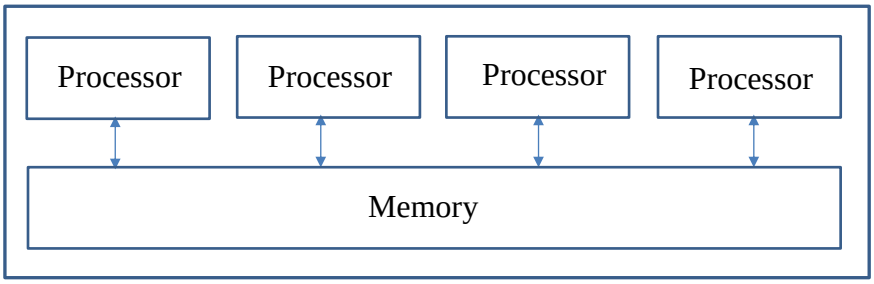


Figure 3.5 Parallel system with shared memory.

Traditional shared memory parallel computing on CPUs by implementing OpenMP (Open Multiple Processing) has been used for past two decades which is briefly reviewed in Section 3.3. Recently, General-purpose computing on graphics processing units (GPU) which takes advantage of hundreds of cores on GPUs became popular for scientific purposes in less than a decade. GPU parallel computing can deliver performance that is often in the order of magnitude faster the same computation on a multi-core CPU (Brodtkorb *et al.*, 2012). The implementation of the heterogeneous parallel computing technique using both CPU and GPUs for the proposed hydrodynamic tsunami model is explained in details in Section 3.4.

The ideal speed up of a parallel model compared to its sequential version is expected to increase in proportion to the number of processors used. However, the speedup of an algorithm on a parallel computing platform is limited by the serial portion of the problem governed by Amdahl's law (Amdahl, 1967). This law states that the maximum speedup (S) of a program as following:

$$S = \frac{1}{(1-P) + \frac{P}{N}} \tag{3.36}$$

Here P is the fraction of the total serial execution which can be parallelized and N is the number of processors. For example if 90 % of an algorithm is parallelizable the computational time of the parallel model running on a quad-core CPU expected to be reduced by about three times compared to the sequential version which employs a single core of the CPU. In the next chapter the computational performance of the parallelized model on CPU (using OpenMP) and GPU (using CUDA) will be presented.

3.3 OpenMP parallel computing on CPU

OpenMP (Open Multiple Processing) is an Application Processing Interface (API) that contains a collection of compiler directives, library routines and environment variables for shared memory parallel programming with C, C++ or Fortran in most operating systems including Windows and Linux, etc.

The OpenMP paradigm is generally focused on performing loop iterations in parallel (Chandra, 2001). In fact, a programmer should mark the loops in a sequential program which can be parallelized. The compiler directives are mainly written for the main functions from the sequential program section.

In this study, OpenMP 3.1 is used which was released on July 9, 2011, and is implemented by the Visual Studio environment. The original C++ code is directly parallelised by applying several OpenMP directives, which is easier than heterogeneous parallel computing based on both CPU and GPUs. Although this application is rather easy, there are common mistakes in using OpenMP which can be avoided. For example forgetting to declare certain variables as private when they are needed to be used as private. Another mistake is related to not using the OpenMP constructs such as the 'critical', 'atomic' and 'locks' in the regions when they are necessary to be implemented in order to prevent concurrent access to the same memory location. Otherwise, OpenMP-compilers available today are not able to protect the programmer from making these mistakes. Some of these mistakes are categorized, and explained how to avoid them in (Süß and Leopold, 2008).

The functions that can be executed in parallel in the current 2D tsunami model include: the friction source term discretisation, the first and second-order accuracy numerical computation, and updating flow variables for the new time step. As an example, a parallel function within the code using OpenMP directive and clauses can be shown as follows:

```
void function name () {
local variables

#pragma omp parallel for private (variables)

Code block
}
```

OpenMP using the portable and scalable model provides programmers a flexible interface for developing parallel applications for different platforms from the standard desktop computers to the super computers.

3.4 GPU parallel computing and CUDA implementation

GPU-accelerated computing as discussed in the literature is the use of a graphics processing unit (GPU) together with a CPU to accelerate scientific computing codes. GPUs typically have an architecture which is significantly different from traditional CPUs. The architecture of a single CPU and GPU are shown in Figure 3.6.

A CPU consists of a few cores optimized for sequential serial processing while a GPU has a massively parallel architecture consisting of thousands of smaller, efficient cores designed for handling multiple tasks simultaneously. If these thousands of threads can execute in parallel, a tremendous amount of computational power with high memory bandwidth can be leveraged; however, this requires an efficient programme code (NVIDIA, 2012c).

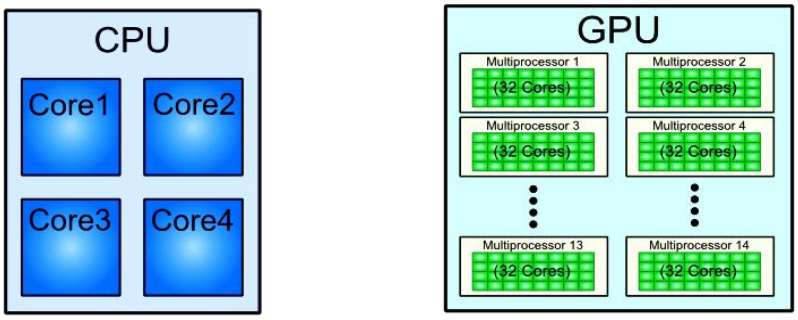


Figure 3.6 Core comparison between a CPU and a GPU.

In this section, the implementation of the aforementioned finite-volume Godunov-type SWE model on GPUs for parallel processing using CUDA is explained. It should be added that the main materials discussed for the rest of this chapter can be found in (Amouzgar *et al.*, 2014; Amouzgar *et al.*, 2016).

Although only compatible with NVIDIA devices, the CUDA platform is accessible to software developers through extensions to industry-standard programming languages, including C, C++ and Fortran, and so is adopted for this work. The code developed herein uses CUDA/C for the wholly parallelised component on GPUs (NVIDIA, 2012b; NVIDIA, 2012c) and C++ for sequential programming on CPUs. The general form for GPU programming comprises sequential and parallel parts, which are illustrated in Figure 3.7. The program begins by allocating memory on the host (CPU) and the device (GPU), followed by loading the required data onto the host. Data read at this point include bathymetry/topography, initial conditions of water level and discharges, and boundary conditions as inflows and outflows during the simulation if necessary. The data allocated within the random access memory (RAM) accessible by the CPU are copied to the global memory (within the dynamic RAM (DRAM)) of the GPU. Parallelised components of the code are defined as kernel functions, written in CUDA/C. Data residing in the GPU memory now can be accessed by the kernel functions for executions. If the end simulation time is reached, or results are required for monitoring or visualisation, the cell state data are similarly copied from GPU to CPU as a contiguous block of memory.

In CUDA the execution model is analogous with a grid divided into a number of blocks; blocks are divided into a number of threads, as illustrated in Figure 3.8. These blocks can be executed

out of order to allow scalability for a varying number of cores in a particular device, whereas threads in a block are executed together in groups of 32 'warps'.

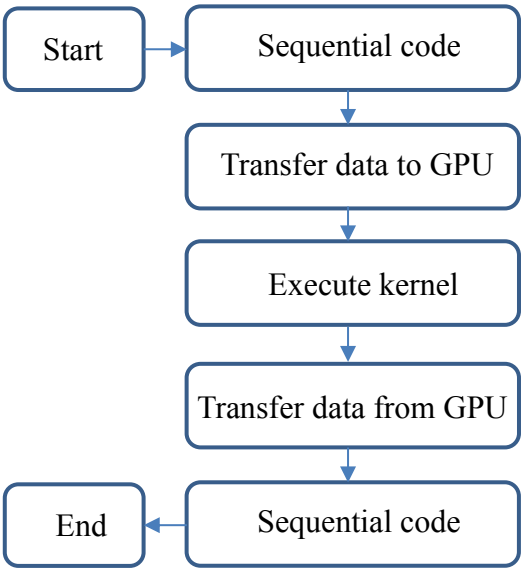


Figure 3.7 Heterogeneous parallel programming diagram.

3.4.1 *Domain structure*

The domain structure applied herein is based on a 2D regular uniform grid of $N \times M$ cells. Although this is a 2D domain, the domain and the relevant datasets are considered as one dimensional (1D) to ensure that the data are allocated and addressed as a contiguous block of memory. It should be mentioned that applying up to three dimensions for the block and thread hierarchy is possible in CUDA, but a 1D array of size $N \times M$ with index element at $[x, y]$ as $x \times N + y$ is adopted in this work. Conversion of a 2D array to 1D as a vector is shown in Figure 3.9.

Figure 3.9 shows an example in which a 4×4 matrix is converted to a vector with 16 elements. The input data such as bathymetry/topography, and initialization are read in the programme as usual in a 2D form.

Then it is transformed to 1D, and a pointer is assigned to the vector which points to the whole data set in that vector. For example if the bathymetry/topography array is assumed as $z_b\,(i, j)$, the following procedure will be implemented in the code:

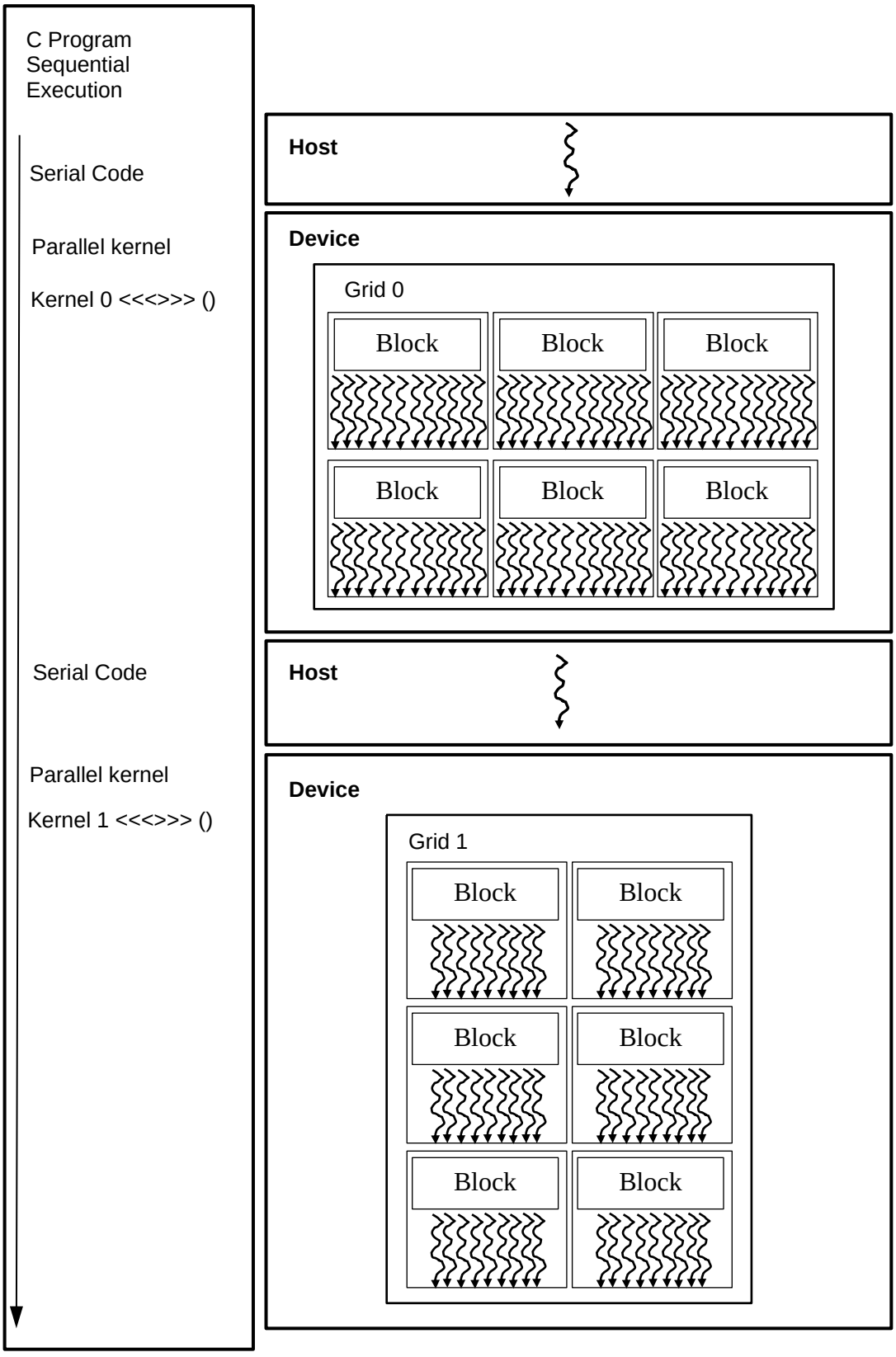


Figure 3.8 Heterogeneous parallel programming structure.

```
// (allocating memory for zb on CPU)
zb =new double [N*M];

// (pointing to the vector zb)
zb [i+j*N] → *zb;

// (allocating memory for zb on  GPU)
cudaMalloc((void**) &zb_device, N*M*sizeof(double));

// Copy data from CPU memory to GPU memory
cudaMemcpy(zb_device, zb, N*M*sizeof(double), cudaMemcpyHostToDevice);
```

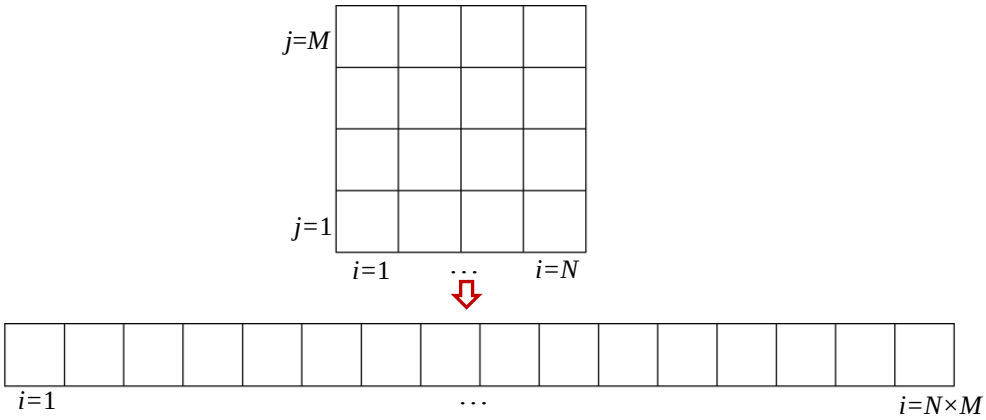


Figure 3.9 Transferring 2D array to 1D.

3.4.2 *Kernel structure*

The main functions which can be parallelised should be defined first. For example for the first-order scheme used in this work, three kernels are developed, including the finite volume solver, the friction discretisation step and the reduction step for time advancement, which will be executed as the main functions on the GPU. For each kernel, a grid of thread blocks should be launched, and it may be written in the following form (Kirk and Wen-mei, 2012; Sanders and Kandrot, 2012):

```
ThreadsPerBlock =Multiple of warp size (i.e. 64);
BlocksPerGrid =(N*M+threadsPerBlock-1)/threadsPerBlock;
Kernel<<<blocksPerGrid, threadsPerBlock>>>;
```

while inside a kernel the thread index would be defined as

```
int index = blockDim.x * blockIdx.x + threadIdx.x;
```

and if the index of a thread exceeds the total number of cells, the kernel is permitted to exit early.

```
__global__ void kernel() {

if (index >= N*M)
{return;}

else
execute the task

}
```

The maximum number of blocks per grid in each dimension and the maximum threads per block are limited to the device capability which should be considered for a proper launching of blocks and threads. For the GPU devices used in this work, the maximum possible number of blocks per grid and threads per block are 65,535 and 512, respectively.

As mentioned above, threads per block should be a multiple of warp size. Occupancy is then defined as the ratio of the number of active warps per multiprocessor to the maximum number of possible active warps (NVIDIA, 2012c). It should be noted that small block sizes prevent high levels of occupancy, whereas large block sizes may consume more resources than are available; in any case scheduling further work once a high occupancy is achieved does not necessarily increase performance. Usually 128–256 threads per block are suggested but this generally depends on the application and device (Sanders and Kandrot, 2012). By checking different values, 64 or 128 threads per block show better performance in this work, which is likely a consequence of the ratio of work to memory transfer, and number of registers required.

3.4.3 *Time-step reduction*

The algorithm for determining the maximum time step allowed is controlled by the Courant–Friedrichs–Lewy (CFL) condition. Its detailed implementation in the context of a Godunov-type scheme and for the numerical stability was discussed in Section 3.1.5. On GPUs, a kernel which iterates through each cell serially would be highly inefficient. There are numerous different algorithms that can be applied for the parallelisation of reductions, fully utilising the power of the processing units. Herein an easy technique is applied, which is available within the thrust library (CUDA Toolkit, 2013). Thrust is a C++ template library for CUDA based on the Standard Template Library (STL). Thrust allows implementation of high-performance parallel applications with minimal programming effort through a high-level interface that is fully interoperable with CUDA/C. This library provides a collection of algorithms and data structures such as scan, sort and reduction in a single package, which is self-contained and requires no additional libraries. For example, to find the permissible time step (`dt`) from the vector (`dti`), the following instruction may be used:

```
dt = thrust::reduce(thrust::device_ptr <double> (dti), thrust::device_ptr <double>
dti + M*N),(double) 3e35, thrust::minimum<double>());
```

3.4.4 *Floating point precision effects*

It is not surprising to find that single precision (32-bit floating point arithmetic) are quicker than double precision (64-bit) because of the different number of floating point operations per second (flops) a device can handle for these two precisions.

Based on experience, single precision can be up to two times faster than double precision. However, there could be significant differences in the results; Smith and Liang (2013) showed that single precision may introduce large errors in mass conservation for the given numerical scheme for a dam-break test case.

Therefore results presented in this work are for 64-bit simulations. However, whether 64-bit computation can be used or not depends on the device's computing capability, which must be version 2.0 or greater. The technical specifications of NVIDIA GPUs based on their computing capabilities can be found in NVIDIA (2012c).

3.4.5 *Model structure*

The diagram of the hydrodynamic model implemented on the GPU is illustrated in Figure 3.10. In the fully parallelized calculation component for the second-order accurate finite volume scheme, four main kernels are defined, including MUSCL-Hancock predictor (half-time step kernel), MUSCL-Hancock corrector (full-time step kernel), friction step and time step reduction which are fully executed on the GPU. The main operations within each kernel are listed as below:

Friction step Kernel

- Friction terms are solved by using an implicit solver
- The wet and dry cells are also categorised

Half-time step kernel for prediction of intermediate cell states

- The slopes are calculated by implementing slope limiters in the *x*- and *y*- directions
- The cell data are linearly reconstructed to the cell faces
- The flux vectors are calculated by the reconstructed cell data
- Half-time evolution of cell state data
- MUSCL reconstruction of face data based on predicting data

Full-time step kernel for advancement of cell states

- Non-negative reconstruction of face data at each cell interface
- Prevent draining flow from a dry cell and locally modify the bed elevation
- Calculate interface fluxes using the HLLC Riemann solver
- Update the flow variables to the new time step

Time step advancement kernel and time step reduction

- The cell-centre flow variables are updated
- The time step vector is determined based on the grid size and the velocities for each cell in the domain by CFL criterion.
- The minimum value of the vector which is the maximum permissible time step will be determined by reduction algorithm within thrust library

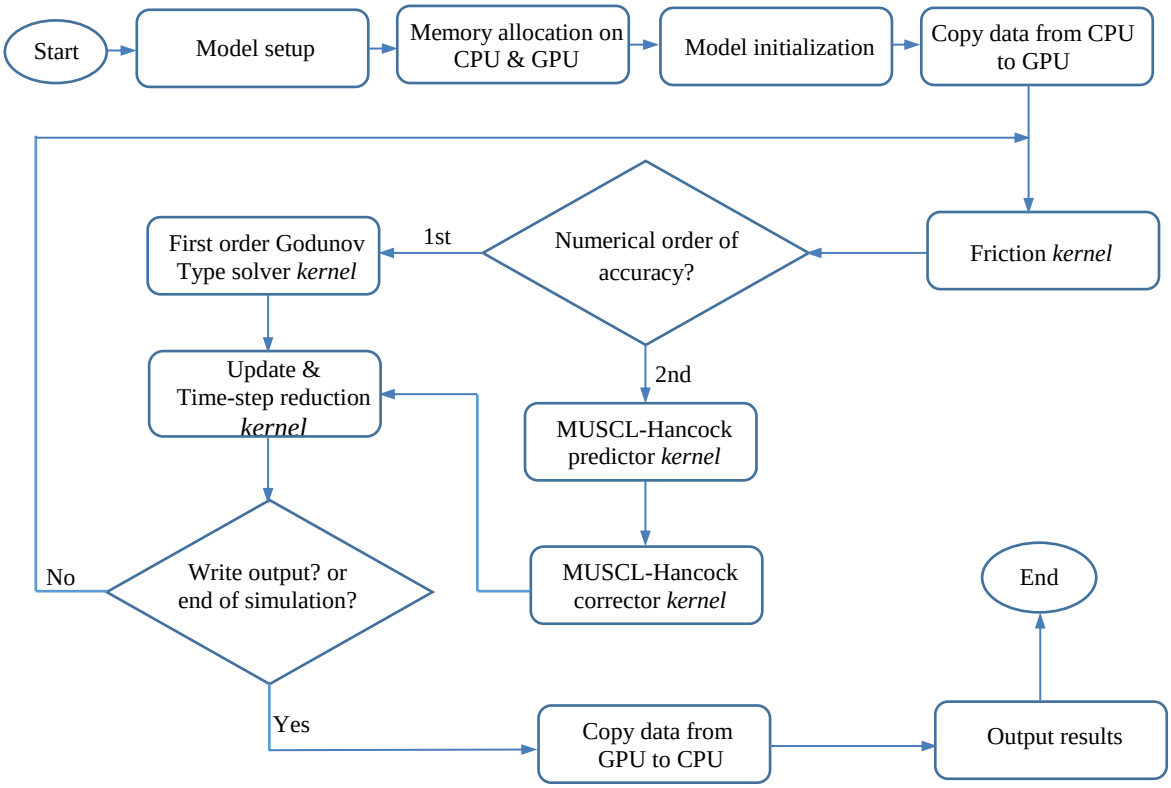


Figure 3.10 Flowchart of the GPU hydrodynamic tsunami model.

3.5 Summary

This chapter has introduced a new hydrodynamic model based on GPU parallel computing for tsunami simulations. The model solves the 2D non-linear SWEs using a MUSCL-Hancock second-order finite-volume Godunov-type scheme incorporated with an HLLC approximate Riemann solver for flux calculations. The numerical techniques adopted to ensure well-balanced non-negative solutions for applications involving wetting and drying over irregular domain bathymetry/topographies are explained. Typical open boundary conditions are implemented for simulation of real world tsunamis, including a simple numerical technique to prevent spurious oscillations and numerical instability caused by the irregular change of bathymetry at the boundary region. To enable simulations on different devices, the numerical scheme has been coded in C/C++ for sequential simulations on a single CPU core and in CUDA

for execution on NVIDIA GPUs. The sequential program has also been improved using OpenMP to allow memory shared parallel computing to take the advantage provided by the modern multiple-core desktop computers. The numerical models discussed in this chapter will be validated against a number of test cases, including in-depth discussion of model performance, in the next chapter.

Chapter 4 Model Validation and Performance

The previous chapter introduced the proposed hydrodynamic model for tsunami simulations and explained the computational acceleration techniques implemented to improve the model performance. In this chapter the tsunami models accelerated by different techniques are validated through various test cases, and the numerical results are evaluated by comparing with analytic solutions, experimental data, field measurements or numerical predictions reported in the literature whenever available. Part of results presented in this chapter can be also found in (Amouzgar and Liang, 2014; Amouzgar *et al.*, 2014; Amouzgar *et al.*, 2016).

4.1 Test cases

The test cases being considered include the circular dam break, dam break over irregular topography, 2D runup of a solitary wave over a conical island involving both non-breaking and breaking waves, laboratory-scale tsunami propagation and run-up at Monai, and the field-scale Okoshiri tsunami generated by the Hokkaido earthquake. The performance of the GPU parallel model will be also discussed through comparison to its sequential CPU counterparts. For certain test cases, the performance across different hardware devices is also compared and discussed. It should be noted that the comparison of runtimes is only indicative as different computer languages may involve different optimization strategies for simulations, although the numerical schemes are identical for both of the CPU and GPU implementations.

The hardware devices used for this research include a four-core Intel Core i5-2500 @ 3.3 GHZ PC, a desktop Geforce GTX 560 Ti GPU and a Tesla M2075 GPU running on a server, as illustrated in Figure 4.1. The specifications of these hardware devices are listed in Table 4.1 and Table 4.2.

(a) (b) (c)

Figure 4.1 Hardware devices used for the simulations: (a) CPU Intel Core i5-2500; (b) GPU GTX 560 Ti; (c) Tesla M2075 GPU on a server.

Microsoft visual studio 2013 is used as an Integrated Development Environment (IDE) to develop the models in C/C++, and CUDA 7.5 is installed to enable the IDE for CUDA

implementation in heterogeneous parallel programming. The windows and Linux operating systems are used for the desktop and server GPUs, respectively.

Processor number	i5-2500
Number of cores	4
Intel smart Cache(MB)	6
Processor base frequency(GHZ)	2.5

Table 4.1 CPU specifications.

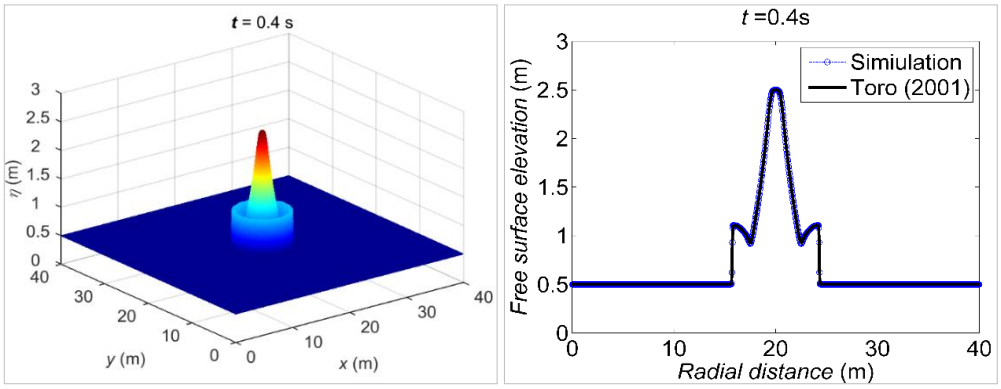
GPUs	Geforce GTX 560 Ti	Tesla M2075
Cores	384	448
Processor clock(MHZ)	1645	575
GPU RAM(GB)	1	6
Bandwidth (GB/s)	128	150
Compute capability	2.1	2.0

Table 4.2 GPU specifications.

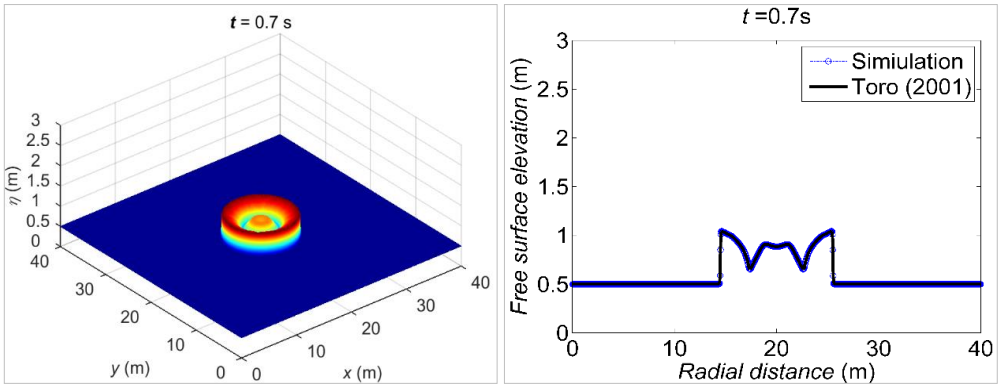
4.1.1 *Circular dam-break*

An idealized test case of circular dam break proposed by Toro (2001) is considered here. In a square domain (40 m × 40 m) with four open boundaries and a frictionless flat bed, an infinitesimal thin circular dam of radius 2.5 m located at the centre divides the domain into two parts that contain respectively 2.5 m and 0.5 m of still water. At $t = 0$, the dam is instantaneously removed to generate a dam-break wave. The computational domain is discretized using a uniform grid of 1000×1000 cells for the simulations.

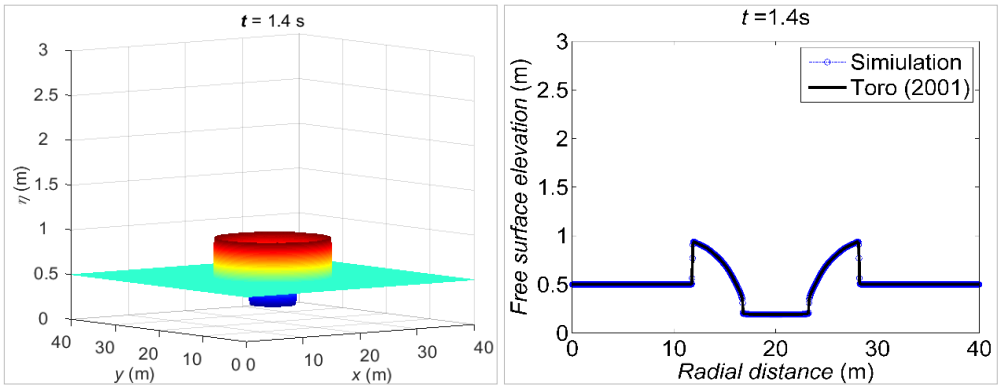
The propagating dam-break wave profiles at t = 0.4 s, 0.7 s, 1.4 s and 4.7 s are presented in Figure 4.2. At t = 0.4 s, a central wave-pattern has evidently formed, consisting of an outward-propagating circular shock wave and an inward–propagating rarefaction wave. At t = 0.7 s the circular rarefaction has imploded into the centre and started to propagate outwards, causing a small dip right at the centre. At t = 1.4 s the circular shock has propagated further in the outward direction; the reflected rarefaction has expanded the flow such that the free-surface level falls below the smaller initial depth outside the dam; the free-surface is close to zero and a secondary shock has formed. At t = 4.7 s, the first shock wave has nearly reached the outer domain boundaries shortly after the secondary shock has reached the domain centre and started to propagate outwards. The numerical results are in excellent agreement with those predicted by other finite volume Godunov-type schemes reported in literature, e.g. (Toro, 2001). Successful reproduction of this test case confirms the shock-capturing capability of the current numerical scheme, which is therefore suitable for simulating the propagating wave front in a tsunami event.



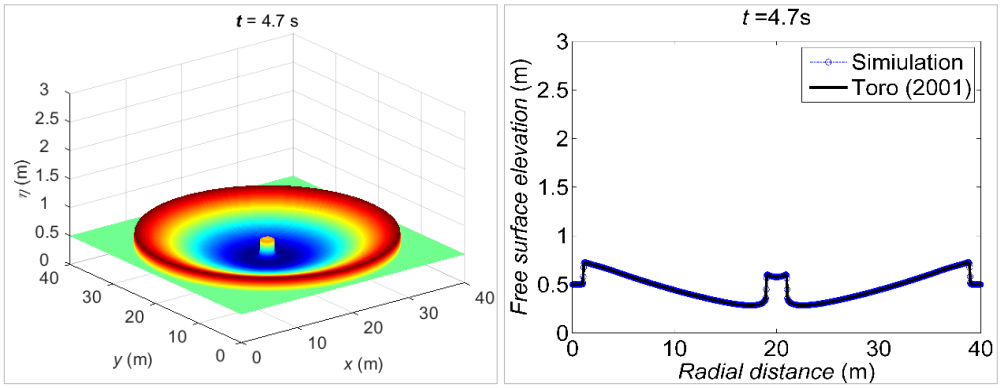
(a)



(b)



(c)



(d)

Figure 4.2 Circular dam-break: left column: 3D water surfaces, right column: corresponding central water surface profiles at (a) t = 0.4 s; (b) t = 0.7 s; (c) t = 1.4 s; (d) t = 4.7 s.

The benefit of adopting GPU heterogeneous computing is demonstrated in Table 4.3, where the runtimes required by different simulations of this 4.7 s circular dam-break test are presented.

CPU Intel core i5-2500	GPU Tesla M2075	Speedup
401.72 s	10.23 s	39.27×

Table 4.3 Circular dam-break: run-times required by different simulations.

From the table, it is evident that the simulation on an NVIDIA Tesla M2075 GPU reduces the runtime about 40 times in comparison to the simulation on a single CPU core, leading to exactly the same numerical solution.

4.1.2 *Dam-break wave interacting with three humps*

This test case considers a dam-break wave propagating over an initially dry floodplain with three humps, which was proposed by Kawahara and Umetsu (1986) to validate their finite element two-dimensional SWE model. This is an ideal benchmark to check the model capability for simulating complex flow hydrodynamics over uneven topography with wetting and drying. The dam break occurs in a 75 m × 30 m closed container, with the bed topography defined as:

$$z_b = \max\left(0, 1-\frac{1}{8}\sqrt{(x-30)^2+(y-6)^2}, 1-\frac{1}{8}\sqrt{(x-30)^2+(y-24)^2}, 3-\frac{3}{10}\sqrt{(x-47.5)^2+(y-15)^2}\right) \tag{4.1}$$

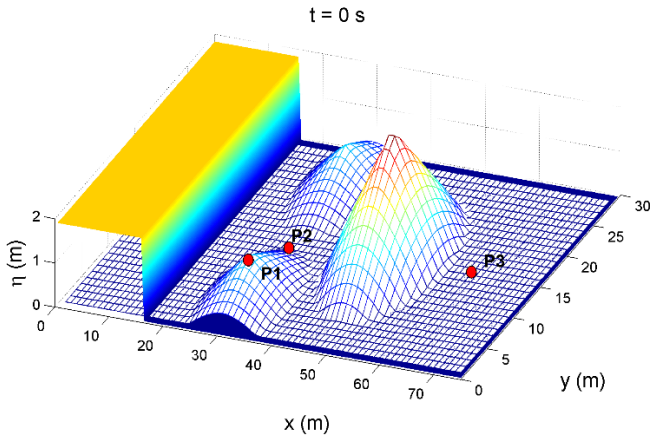
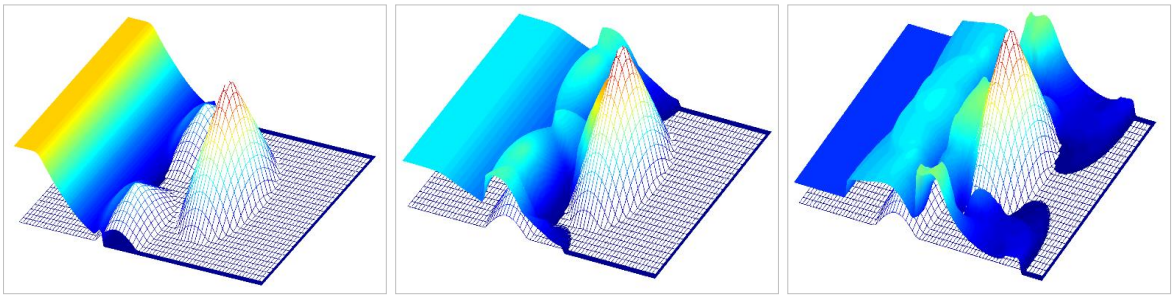
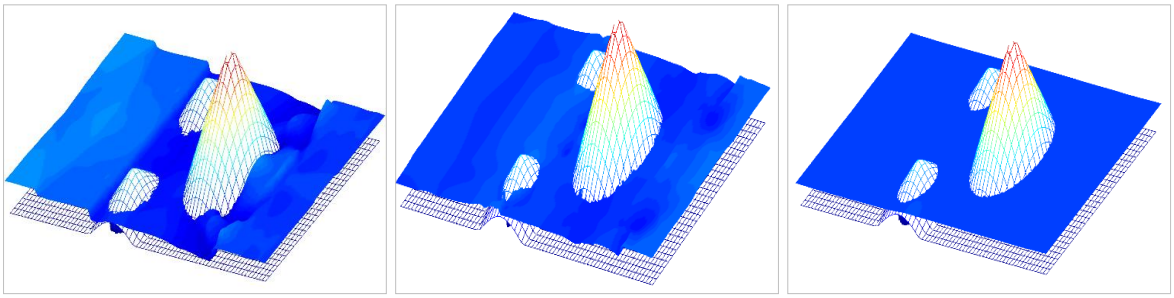


Figure 4.3 Dam-break over three humps: initial conditions and bed topography.

The dam is located at x = 16 m, upstream of which the initial still water level stays at η = 1.875 m while the downstream floodplain is originally dry. The bed topography and initial state of the problem is illustrated in Figure 4.3, where the locations of the three gauge points for recording time histories of flow variables are also indicated. The bed roughness is represented by a constant Manning coefficient of n = 0.018 s/m$^{1/3}$ for the whole domain. The uniform grid of 0.1 m resolution is used during the simulations.



(a) t = 2 s (b) t = 6 s (c) t = 12 s



(d) t = 30 s (e) t = 60 s (f) t = 300

Figure 4.4 Dam-break over three humps: 3D water surface elevation at (a) t = 2 s; (b) t = 6 s; (c) t = 12 s; (d) t = 30 s; (e) t = 60 s; (f) t = 300 s.

Figure 4.4 demonstrates the visualization of the floodplain inundation using 3D water surface plots at different output times until the flow reaches the final steady state. After the dam fails, the dam-break wet/dry front immediately propagates towards downstream and inundates the floodplain. By t = 2 s the flow has started to climb up the small hills, forming the reflected curved waves that move rapidly upstream. At t = 6 s the small hills have been entirely submerged. The wave front has risen more than halfway up the larger hill with part of the front moving around it. The curved reflected waves continue to propagate upstream and interact with each other and develop into a straight bore after 12 s. Meanwhile, a second shock wave develops behind the big hill and has also started to move upstream; the initial wave front continues to travel downstream towards the eastern closed boundary. After 30 s, the wet/dry front has reached the eastern boundary and formed another reflected shock travelling backwards; the second reflected shock behind the big hill has arrived at and reflected from the western boundary to downstream. The wave-topography, wave-wave and wave-wall interactions continue to take place and lead to very complex wave patterns until the momentum of the dam break dissipates through bed friction and the final steady state is reached, as shown at 300 s. It is evident that the current GPU tsunami model correctly simulates this hypothetic dam-break event involving a wetting and drying process. The results are very similar to those reported in literature (Brufau *et al.*, 2002; Liang and Borthwick, 2009; Kesserwani and Liang, 2010).

To further demonstrate the performance of the current model, simulations of different resolutions and different orders of numerical accuracy are performed and the predicted results

are compared, in terms of time histories of water depth recorded at the three gauge points, P1, P2, and P3 which are located at (30 m, 6 m), (30 m, 15 m), and (65 m, 15 m), respectively, as illustrated in Figure 4.3.

Figure 4.5 presents the time histories of water depth at the three gauge points predicted on uniform computational grids of different resolutions. The results are generally consistent, showing a trend of convergence across resolutions, from low to high.

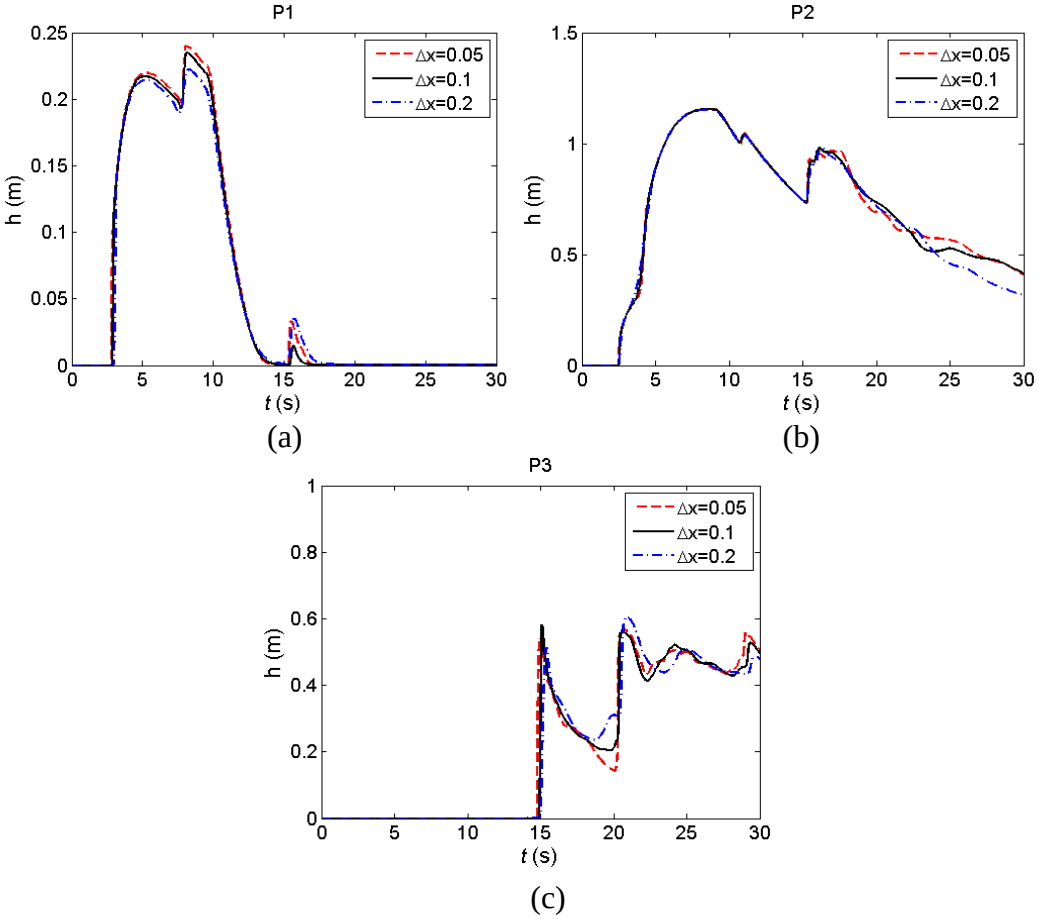


Figure 4.5 Dam break over three humps: water depth histories predicted by the 2nd-order numerical scheme at different resolutions at the three gauge points, (a) P1; (b) P2; (c) P3.

Figure 4.6 compares the simulation results obtained using the 1st and 2nd-order numerical schemes at the spatial resolution of 0.1 m. The results clearly demonstrate the advantage of the 2nd-order scheme in better resolving the sharp-fronted flow hydrodynamics with less numerical dissipation. This essentially indicates that results of similar accuracy may be theoretically obtained using a high-order scheme at relatively lower spatial resolutions. Considering tsunami waves are typically characterized as wall of water rapidly moving inland over a vast domain, higher, at least 2nd-order, numerical scheme may be essential for reliable predictions.

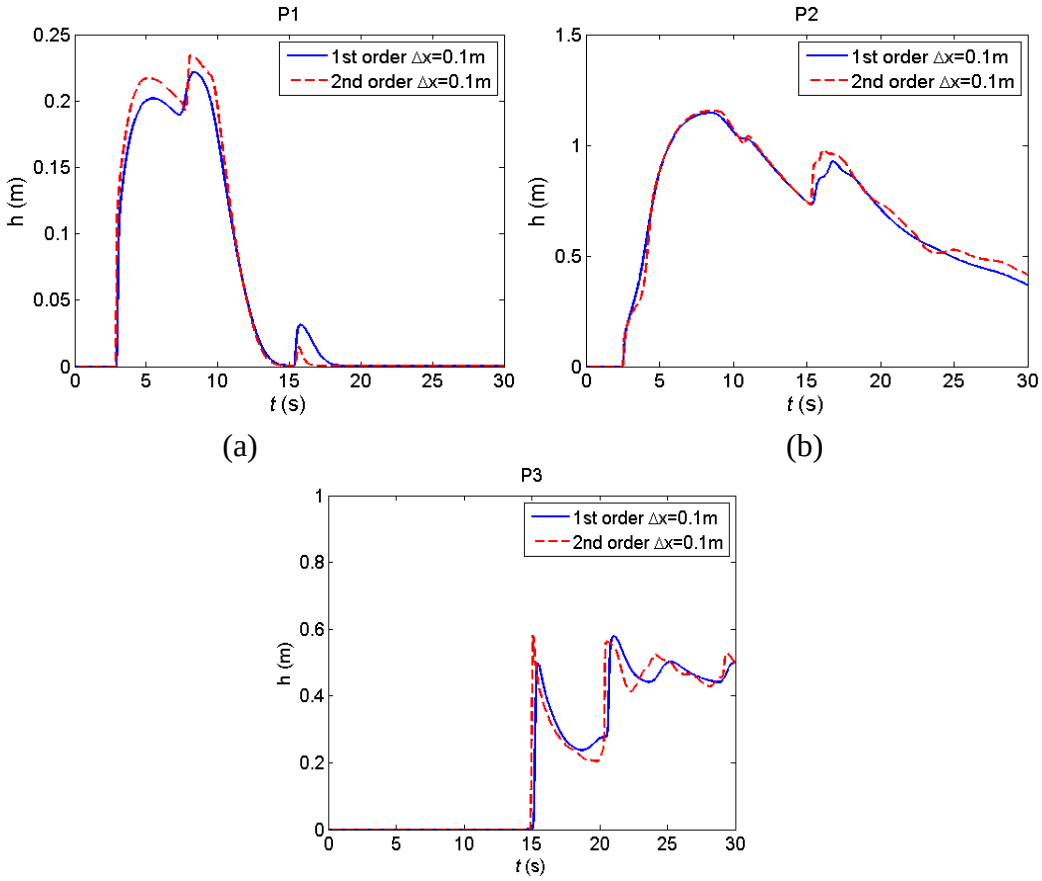


Figure 4.6 Dam break over three humps: water depth histories predicted by the 1st and 2nd-order accurate numerical schemes at the 0.1m spatial resolution at the three gauge points, (a) P1; (b) P2; (c) P3.

Table 4.4 provides the runtimes required by the current 2nd-order tsunami model on different devices, i.e. a single standard CPU core and a NVIDIA Tesla M2075 GPU. The simulations are run on 0.1 m resolution for a 30s event. The GPU simulation is found to be 44 times faster than that on the CPU.

CPU Intel core i5-2500	GPU Tesla M2075	Speedup
802.53 s	18.29 s	43.88×

Table 4.4 Dam break over three humps: runtimes required by different simulations.

More simulations are then performed to investigate the relationship between the GPU speedup and the number of cells being used and the results are presented in Table 4.5 and plotted in Figure 4.7. The performance of the GPU clearly increases with more cells involved in the simulations and the speedup in runtimes gradually reaches its limit after ~1 million of cells are used for the simulations.

Resolution (m)	Number of cells	Speed up (×)
0.3	25,000	31.07
0.25	36,000	32.98
0.2	56,250	38.18
0.15	100,000	42.54
0.1	225,000	43.88
0.075	400,000	47.55
0.05	900,000	51.69
0.025	3,600,000	53.7

Table 4.5 Dam break over three humps: the model speed up with different number of cells in comparison to the corresponding sequential simulation on a single CPU.

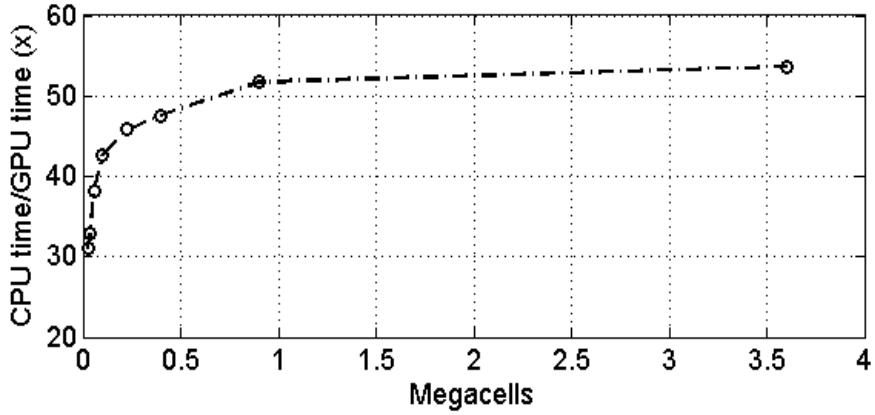


Figure 4.7 Dam break over three humps: GPU speedup against the number of cells being used for simulations.

The increase of performance for solving the problem with larger number of cells may be explained by the fact that, with a large number of cells, more active blocks occupy the GPU cores for longer time and the GPU calculation better balances the CPU-GPU communication overhead. It should be reminded that the speedup performs only a rough estimation of the efficiency since the performance is biased by the CPU's architecture and compiler in use.

4.1.3 *2D run-up of a solitary wave on a conical island*

This experimental benchmark test of tsunami run-up onto a conical island was conducted in the US Army Engineer Waterways Experiment Station (Briggs *et al.*, 1995). It is simulated herein to demonstrate the proposed SWE model's capability for simulating non-breaking and breaking waves and complex flow hydrodynamics with wetting and drying over uneven topography. The experimental setup is illustrated in Figure 4.8, where the conical island, with a base diameter of 7.2 m, top diameter of 2.2 m and height of 0.625 m, is located near the center of a 30 m × 25 m basin. For numerical simulations, the computational domain is set to 25.92 m × 27.6 m with the initial water depth equal to 0.32 m.

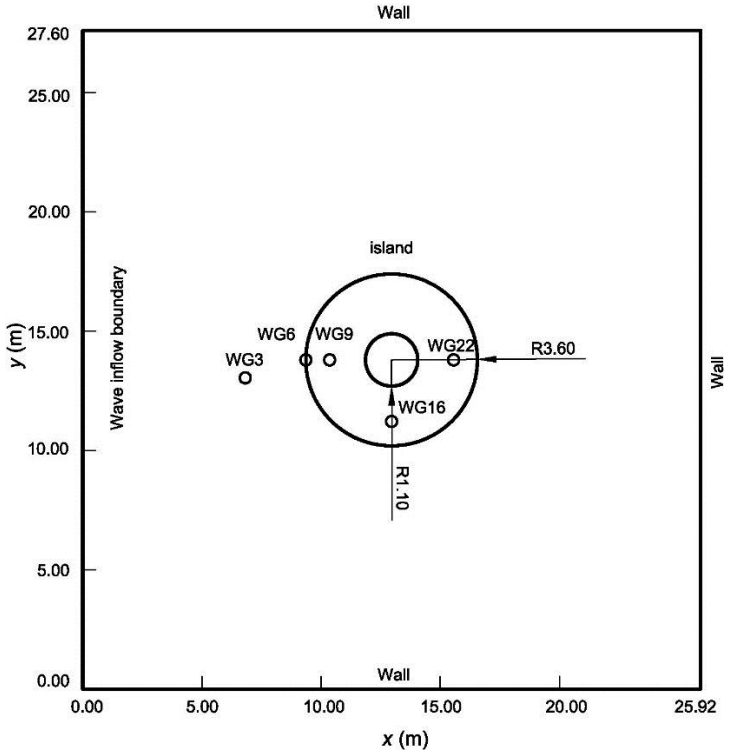


Figure 4.8 Solitary wave run up on a conical island: computational domain, boundary conditions and gauge locations.

The incident wave is imposed from the left boundary at $x = 0$, in order to replicate the solitary wave generated by a wave-maker. The varying wave height z and velocity u are specified as follows

$$z(t) = H\mathrm{sech}^2[\sqrt{\frac{3H}{4D^3}}C(t - T)] \quad , u(t) = \frac{Ch(t)}{D + h(t)}, \quad v(t) = 0 \tag{4.2}$$

where D is the still water depth, H is the wave amplitude, T represents the time when the wave crest reaching the domain and $C = g\ (D + H)^{0.5}$ is the wave celerity. The incident waves with an amplitude of H = 0.032 m and 0.064 m are considered to represent non-breaking (case B) and breaking waves (case C). The still water depth D = 0.32 m and T = 2.45 s are used. Bed friction is neglected based on the findings in Liu *et al.* (1995a), as it was noted that the predicted results are not sensitive to the bed friction due to the steep 1:4 slope of the conical island. The uniform grid resolution is set to 0.04 m in order to be consistent with other works reported in the literature, e.g. Hubbard and Dodd (2002).

Non-breaking wave (case B):

Figure 4.9 presents the simulation results in terms of 3D water surface plots at four sample output times for the simulations on the 0.04 m grids. The incident wave propagating from the left boundary has hit the island and caused runup at the front side of the island at t = 9 s. As the wave run-up reaches the highest level, it starts to retreat back to the initial shoreline. A reflected

wave has also been created and propagates outwards as a circle. The original wave front passes around the island and diffracted waves are developed at t = 12 s. The diffracted wave fronts meet at the lee side and cause runup at the rear side of the island at t = 13 s. After that, the waves continue to propagate further in different directions as observed at t = 14 s.

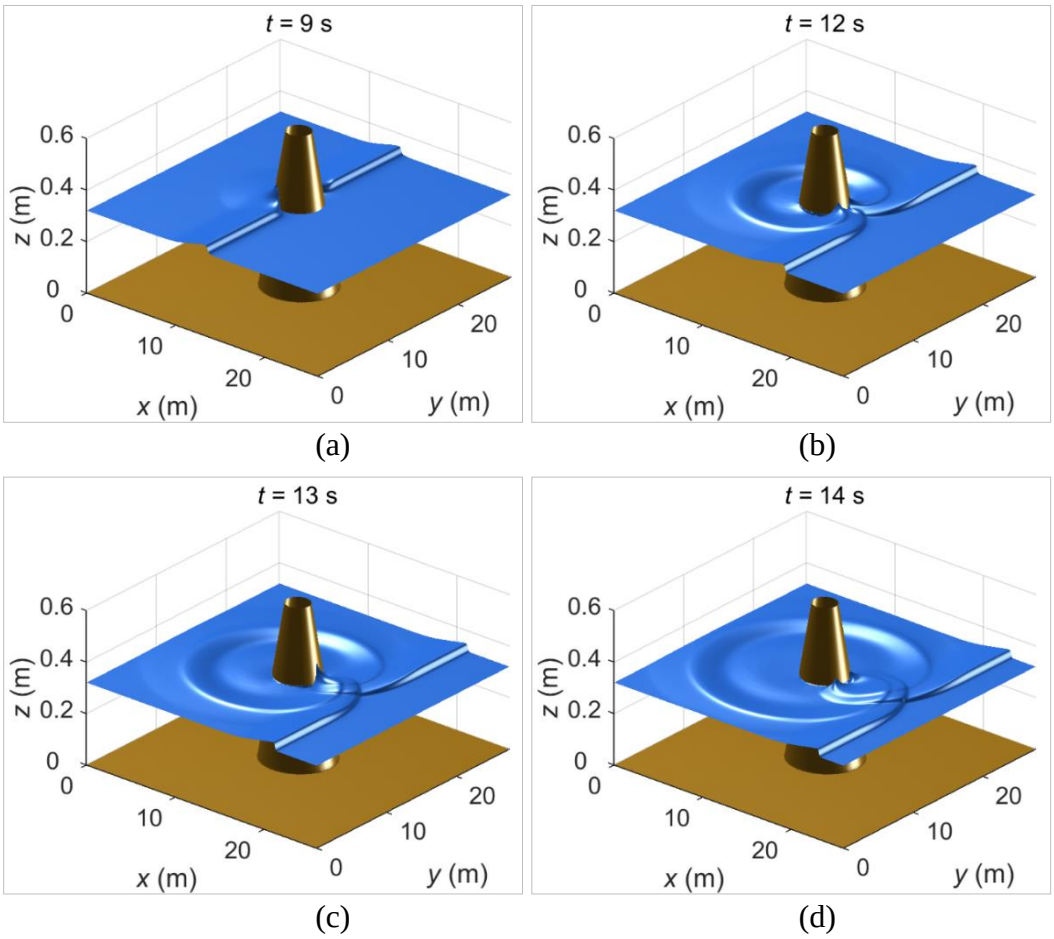


Figure 4.9 Solitary wave run up on a conical island, case B: 3D view of the wave run up at different output times (a) t = 9 s; (b) t = 12 s; (c) t = 13 s; (d) t = 14 s.

To validate the current tsunami model, the numerical predicted time histories of water surface elevation are compared with those recorded at the five gauges G3, G6, G9, G16, and G22 as specified in Table 4.6. The time histories are plotted in Figure 4.10, in comparison with laboratory measurements and alternative numerical results. From the comparisons, it is clear that the present model has done a reasonable job in reproducing the free surface profiles without numerical oscillations. At most gauges, good agreement has been achieved for the lead wave heights and arrival times. The main discrepancies are related to the steepness of the leading wave front that is predicted earlier than in experiment record and the underestimation of the secondary depression wave that follows the lead wave. The main reason for these differences are due to the 3D nature of the complex wave and flow hydrodynamics that cannot be mathematically represented by the SWEs (Choi *et al.*, 2007). Similar discrepancies have been

also reported by other researchers using alternative 2D depth-averaged models (Liu *et al.*, 1995a; Hubbard and Dodd, 2002; Lynett *et al.*, 2002b). The results presented here are in good agreement to those reported in (Hubbard and Dodd, 2002; Nikolos and Delis, 2009; Hou *et al.*, 2013); the results reported by Nikolos and Delis (2009) are also included in Figure 4.10 for comparison.

Wave gauge	*x*	*y*
G3	6.82 m	13.05 m
G6	9.36 m	13.80 m
G9	10.36 m	13.80 m
G16	12.96 m	11.22 m
G22	15.56 m	13.80 m

Table 4.6 Solitary wave run up on a conical island: gauge locations.

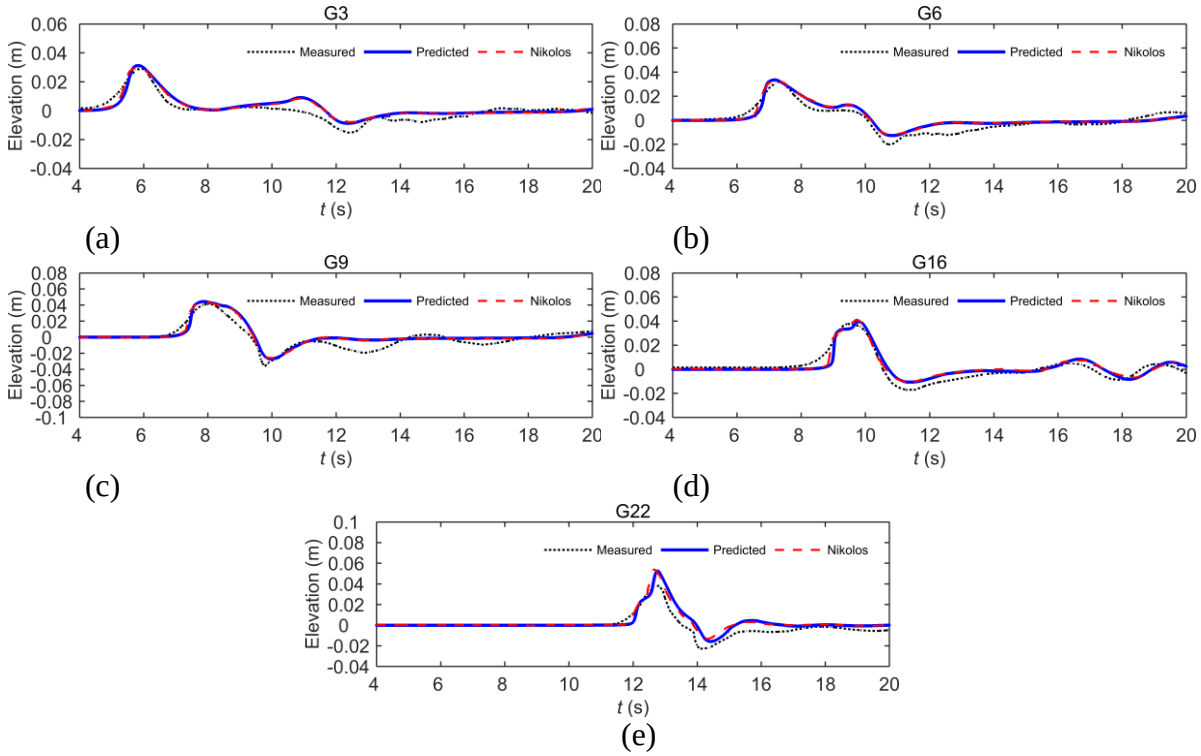
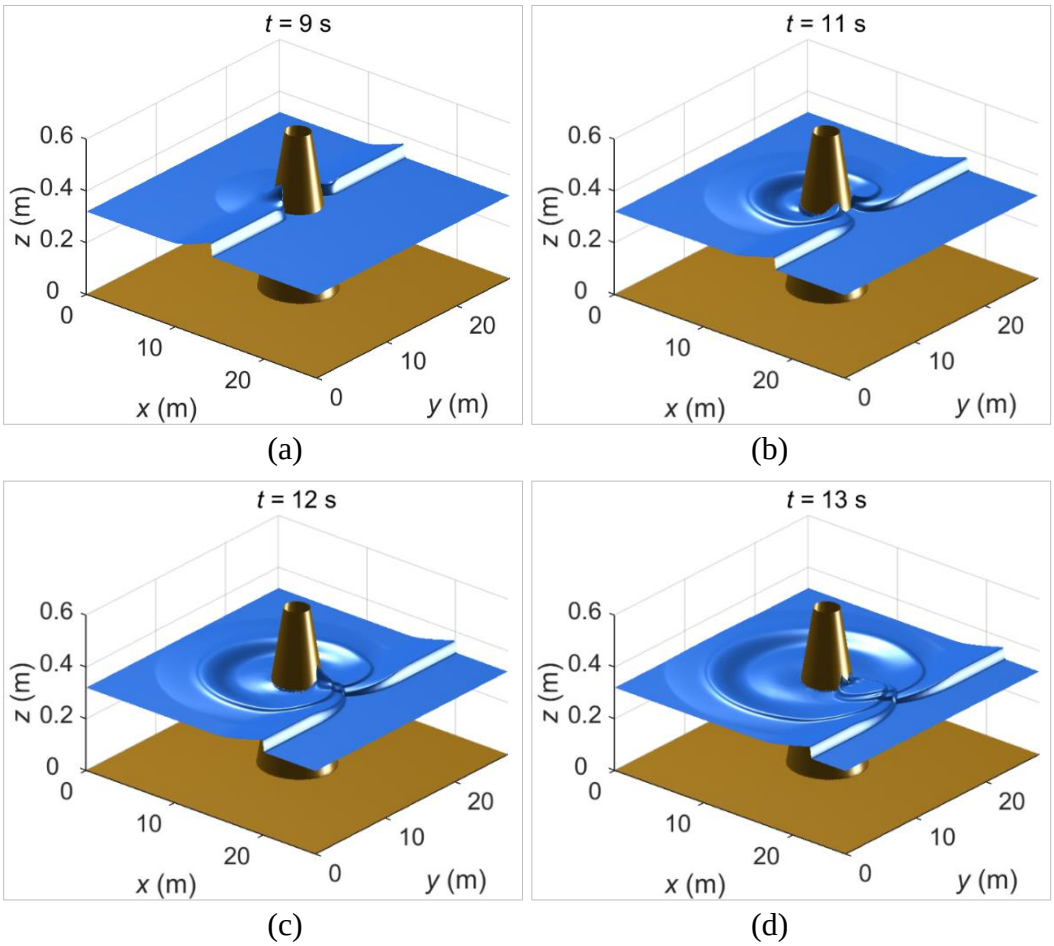


Figure 4.10 Solitary wave run up on a conical island, case B: simulated and measured free surface elevations at five gauges (a) G3; (b) G6; (c) G9; (d) G16; (e) G22.

Breaking wave (case C):

Figure 4.11 presents a series of 3D water surface to show the interaction between the incident solitary wave and the conical island for the case involving breaking waves. The incident wave causes high run-up and inundation at the front side of the island at around t = 9 s. After reaching the maximum run-up, the wave runs down the inundated region and the refracted wave propagates around the island towards the lee side, as shown for t = 11 s. Then these two waves

collide at the lee side producing the second high run-up at about t = 12 s. After that, the waves continue to propagate further in different directions around the island, as observe at t = 13 s. The predicted wave pattern for non-breaking and breaking wave case are similar, however, the arrival time and the magnitude of the wave is different.



(a) (b)

(c) (d)

Figure 4.11 Solitary wave run up on a conical island, case C: 3D view of the wave run up at time (a) t = 9 s; (b) t = 11 s; (c) t = 12 s; (d) t = 13 s.

To further validate the current model for handling this breaking wave case, the predicted time histories of water surface elevation at five gauges are compared with experimental measurements in Figure 4.12. The numerical results agree satisfactorily with the measurements although certain level of discrepancy is also predicted as in the non-breaking wave test. For example, at gauge 3 there is an obvious phase difference between the predicted and recorded leading waves, which is caused by the way the SWEs describing breaking waves. In this case, the physical incident wave breaks before arriving to the shoreline and the SWE model simulates the breaking waves as a propagating bore. The predictions are consistent with numerical predictions reported by other researchers, e.g. Nikolos and Delis (2009) using an unstructured grid based finite volume Godunov-type model implemented with a Roe approximate Riemann solver.

Nevertheless, the arriving time and magnitude of the leading wave are accurately reproduced, which are the most important aspects for engineering considerations. Table 4.7 presents the Root Mean Square Error (RMSE) calculated at the five different gauges, which shows that the average deviation of numerical prediction compared to the measurement is 0.0095 m.

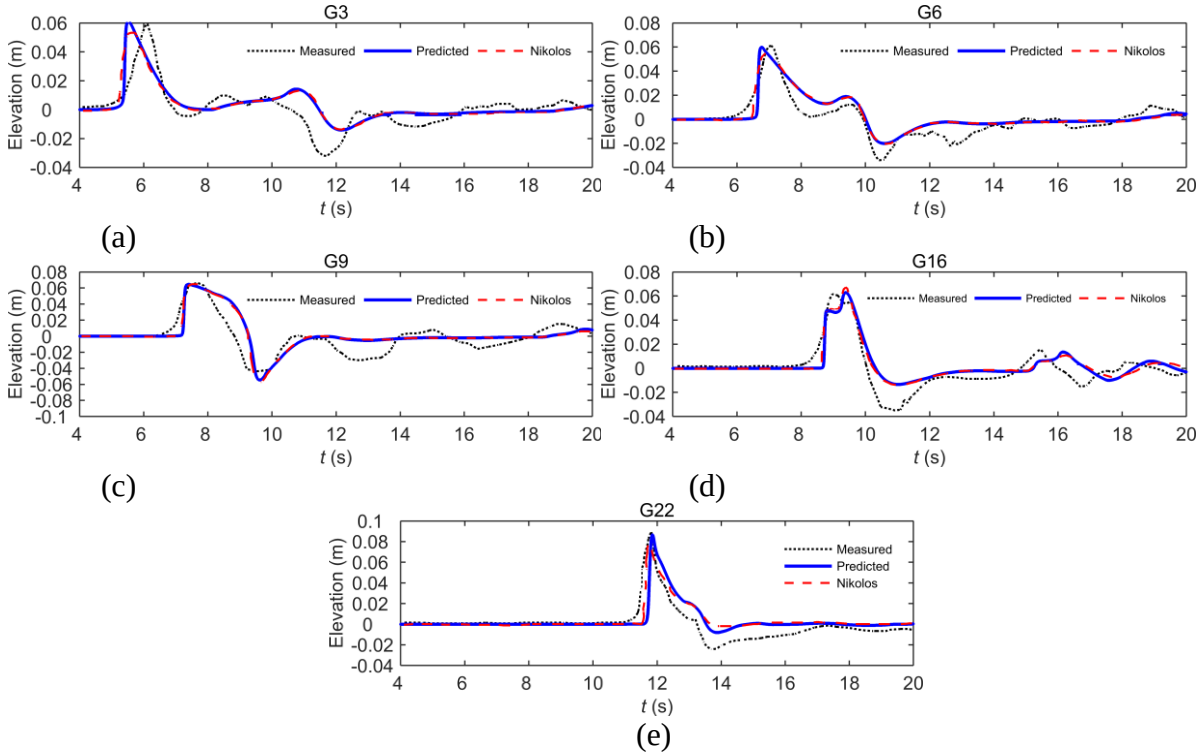


Figure 4.12 Solitary wave run up on a conical island, case C: simulated and measured free surface elevations at five gauges (a) G3; (b) G6; (c) G9; (d) G16; (e) G22.

Gauge no.	3	6	9	16	22
RMSE (m)	0.0126	0.0092	0.0070	0.0083	0.0102

Table 4.7 Solitary wave run up on a conical island, case C: root mean square error (RMSE) at different gauges.

To demonstrate the effect of the grid resolution on the numerical results, further simulations are run on uniform grids of finer and coarser resolutions, i.e. 0.01 m, 0.02 m, 0.08 m and 0.16 m, respectively. The predicted time histories of water surface elevation predicted by the different simulations are shown in Figure 4.13 for gauge 22 at the lee side of the island, and their corresponding maximum run up presented in Table 4.8. The simulation results appear to be convergent in capturing the peak with increasing grid resolution.

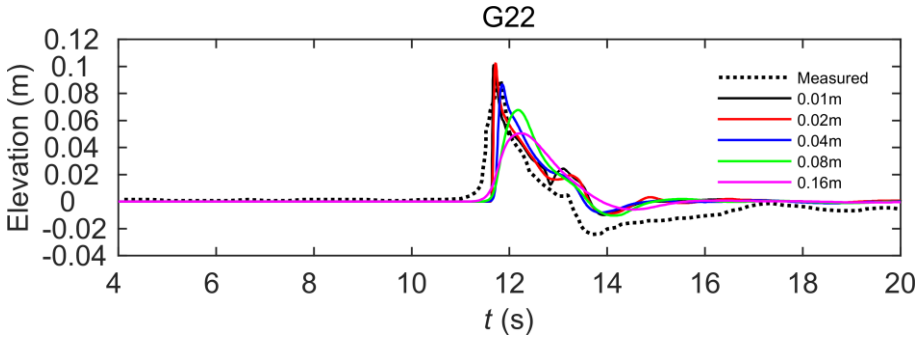


Figure 4.13 Solitary wave run up on a conical island, case C: effect of grid resolution.

Resolution, Δx (m)	0.16	0.08	0.04	0.02	0.015	0.01
Maximum run-up (m)	0.050	0.068	0.087	0.103	0.101	0.101

Table 4.8 Solitary wave run up on a conical island, case C: effect of grid resolution on prediction of maximum run up at gauge 22.

The performance of the current GPU tsunami model is evaluated by comparing the runtimes of different simulations (20 s of simulation with 0.04 m grid resolution and with a coarser resolution of 0.08 m) on different devices. The required computational time for different simulations is presented in Table 4.9. Using the computational time of the single CPU simulation as a reference, the OpenMP memory shared parallelization is shown to gain over three times of computational efficiency and the speedup is consistent across simulations at different resolutions. For the CPU which contains four cores, according to Amdahl's law, less than four times speed up is expected and it is consistent with the performance of the CPU parallel model compared to the sequential version as shown in Table 4.9. Considering that implementation of OpenMP parallelization does not involve major modifications to the codes, it provides a straightforward way to achieve high-performance computing on modern PCs that are normally equipped with multiple cores. Also from Table 4.9, GPUs apparently provides much more potential in terms of improving model performance by substantially reducing the computational cost. The Geforce GTX 560 Ti GPU and the Tesla M2075 GPU are demonstrated to respectively speed up the simulations by 29 times and 36 times on the 0.08 m uniform grid of 111,780 cells.

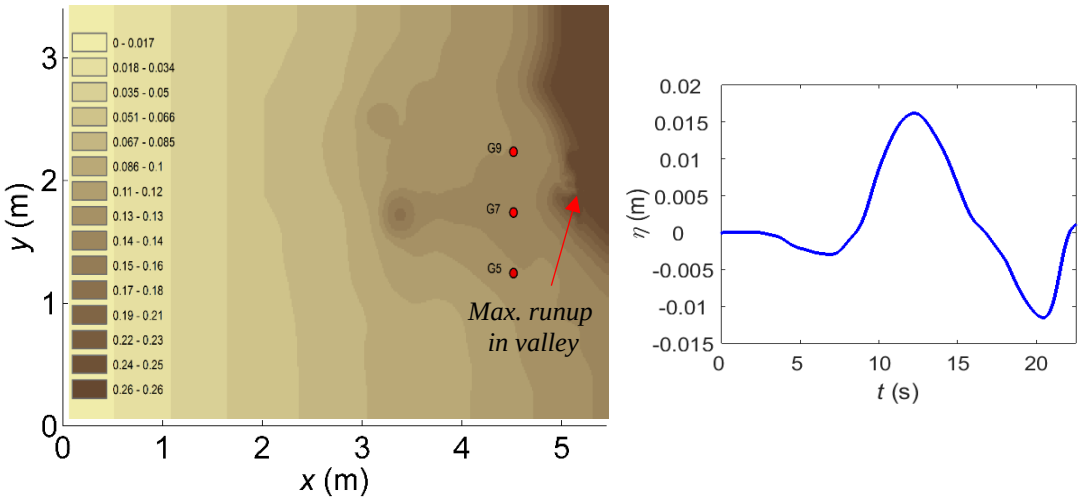
Δx	Number of Cells	CPU	CPU+OpenMP	GPU GTX	GPU Tesla
0.08 m	111,780	106.61 s (~1 ×)	32.31 s (~3.3×)	3.66 s (~29.13×)	2.95 s (~36.14×)
0.04 m	447,120	939.22 s (~1×)	276.24 s (~3.4×)	24.84 s (~37.81×)	21.8 s (~43.08×)

Table 4.9 Solitary wave run up on a conical island: computation time on different hardware devices and the numbers inside the brackets indicate the model speed up in comparison to the corresponding sequential simulation on a single CPU node.

For the higher resolution simulations on the 0.04 m uniform grids of 447,120 cells, the speedup becomes 38 and 43 times, respectively. This indicates that the performance of GPUs increases with increasing computational intensity, which is consistent with the conclusions from our earlier investigations and those reported by other researchers e.g. (Vacondio *et al.*, 2014). Specifically considering the performance of the two GPUs, the Tesla M2075 GPU performs consistently better than the desktop Geforce GTX 560 Ti for both of the simulations. A major reason may be because the Tesla card is equipped with more cores. However, the difference in performance is only 14% and 24% for the high and low resolution simulations, with the Tesla GPU performed better. This indicates that the professional Tesla M2075 GPU does not compare much favourably with the much cheaper desktop GPU, in terms of reducing computational cost. However, for practical applications involving large-scale simulations and massive data processing, the intensive memory requirements make the professional GPUs like Tesla M2075 the only option, supported by their advanced architectural design containing much larger memory.

4.1.4 *Laboratory-Scale Monai Tsunami Benchmark Test*

This tsunami benchmark test aims to reproduce a tank experiment designed to reproduce the 1993 Okushiri tsunami in Japan (Matsuyama and Tanaka, 2001). This tsunami was triggered by a magnitude 7.8 earthquake in the Sea of Japan off southwest Hokkaido, killed over 200 people and caused huge economic losses. The 1 : 400 model was built in a wave flume of 3.4 m wide, 205 m long. In the experimental test, the topography was made relatively complex to represent a narrow valley near the village of Monai as shown in Figure 4.14 (a).



(a) (b)

Figure 4.14 Monai tsunami benchmark: (a) problem domain, (b) incident tsunami wave.

Experimental data were obtained for validating numerical models, including time series at selected locations. Incident tsunami wave, as provided in Figure 4.14 (b), was assumed to enter the domain through the western edge of the 5.448 m × 3.402 m problem domain. Other boundary walls were treated as reflective. During the numerical simulation, the domain is decomposed into uniform grid of 393 × 244 cells that gives a resolution of 0.014 m. A constant Manning coefficient of 0.001 $s/m^{1/3}$ is used in the entire domain as recommended by other researchers (Funke *et al.*, 2011; Popinet, 2011) . The initial still mean water level is 0.13535 m.

Figure 4.15 presents the tsunami propagation and run-up at different outputs times. The first stage of the event involves a retreat of shoreline caused by the initial negative tsunami wave, which is evident in the results shown for t = 12 s. At t = 14 s, the positive wave front has travelled past the island. At t = 16 s, the tsunami has flooded the coastline and the island and is being reflected away from the shore. The maximum run-up occurs after about t = 17 s within the pocket beach as reported by other researchers (LeVeque and George, 2008; Franchello, 2010). The maximum run up height is calculated to be 0.07 m in the Monai valley, which is equivalent to 28 m at the prototype scale and roughly compatible with the field measurement of about 30 m (Matsuyama and Tanaka, 2001). These results appear to be very similar to those obtained by other researchers using alternative models e.g. (LeVeque and George, 2008), (Popinet, 2011) and (Yamazaki *et al.*, 2011a).

Laboratory measurements at three gauge points G5, G7, and G9, which are located at (4.521 m, 1.196 m), (4.521 m, 1.696 m) and (4.521 m, 2.196 m), respectively, are available for comparison with the numerical predictions obtained from the current model in terms of water surface elevation, as illustrated in Figure 4.16. In the first 10 seconds, due to the initial water disturbances in the laboratory wave tank, the predicted water levels do not agree with the measurements. Nevertheless, overall satisfactory agreement has been observed between the numerical results and measurements for both the phase and amplitude of the tsunami wave. The results are consistent with predictions reported by other researchers, e.g. (Zhang and Baptista, 2008; Nicolsky *et al.*, 2011). Predictions from the 1st-order scheme (Amouzgar *et al.*, 2014) are also included and compared, which agree closely with the 2nd-order results, apart from certain minor differences.

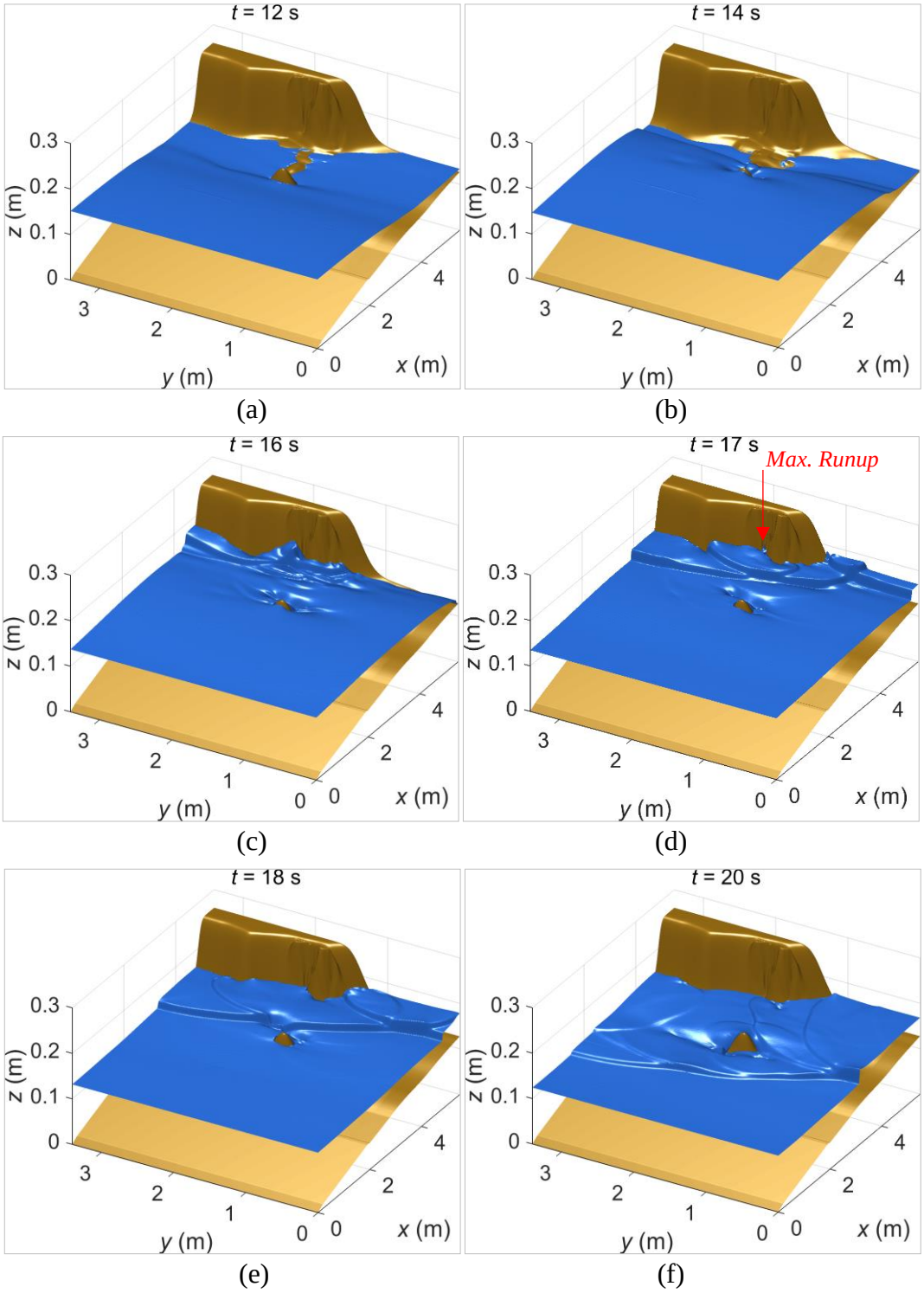


Figure 4.15 Monai tsunami benchmark: three-dimensional view of free surface elevation at time (a) t = 12 s; (b) t = 14 s; (c) t=16 s; (d) t=17 s; (e) t=18 s; (f) t=20 s.

The performance of the current tsunami model is again compared and analysed for simulations considering sequential and parallel computing on different hardware devices. The computational times of a 25 s simulation required by the first and second-order accuracy schemes on different devices considering 0.014 m resolution (94,561 cells) are summarized in Table 4.10. Comparing with the computation on a single CPU core, the simulations on four CPU cores, Geforce GTX 560 Ti GPU and the Tesla M2075 GPU are sped up by 3, 33 and 41

times, respectively, which is generally consistent with the results reported previously for the Conical island benchmark. For the two GPUs, the Tesla card saves 24% of the computational cost and is also consistent with the findings from the previous test case. The GPU model is found to be much more efficient than the traditional CPU code and saves about 41 times of the runtime. This once again confirms the advantage of using GPU parallel computing for high-performance tsunami simulations. Also from the computational runtime presented in the table, it is deduced that the second-order accuracy in time and space is only up to 30% more expensive compared to the first-order accuracy for the numerical scheme used in this work.

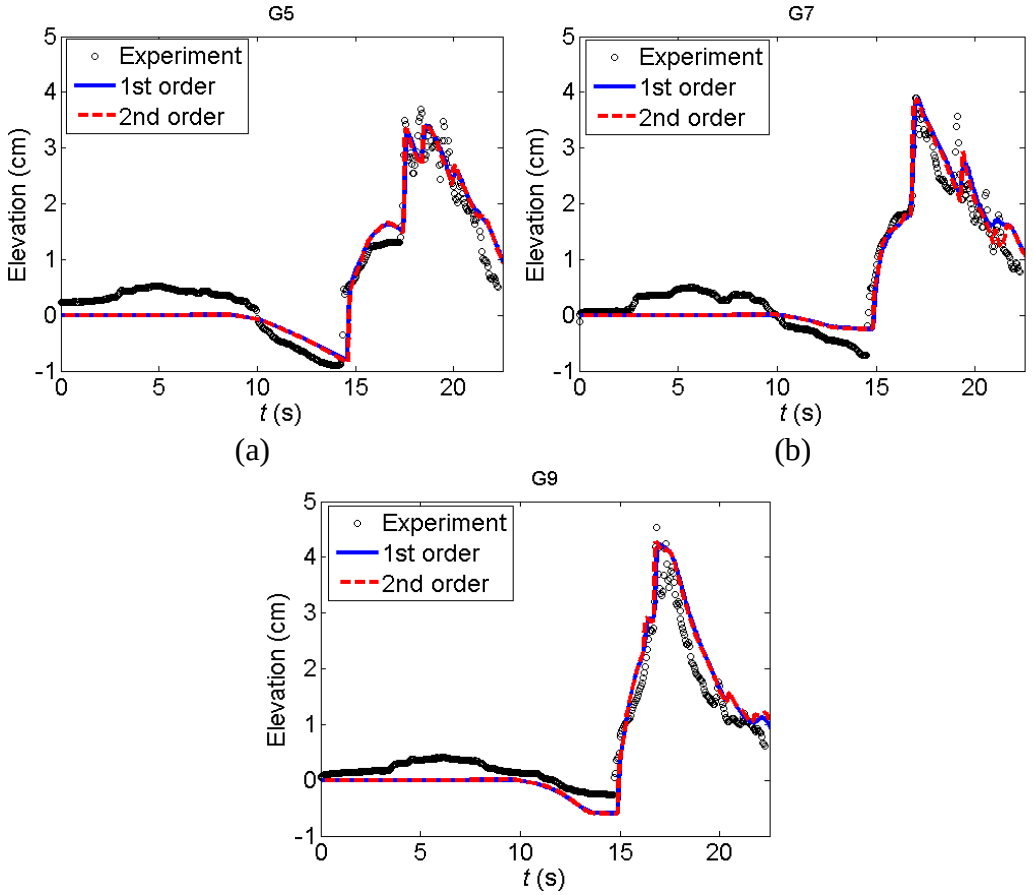


Figure 4.16 Monai tsunami benchmark: 1st and 2nd-order prediction of free surface elevation at three gauges, in comparison with laboratory measurements, at gauges (a) G5; (b) G7; (c) G9.

Order of accuracy	CPU	CPU+OpenMP	GPU GTX	GPU Tesla
1st	371 s (~1×)	115.9 s (~3.2×)	11.04 s (~33.6 ×)	8.9 s (~41.7×)
2nd	472.18 s (~1×)	147.6 s (~3.2×)	14.21 s (~33.23×)	11.46 s (~41.2×)

Table 4.10 Monai tsunami benchmark: computation time on different hardware devices (the numbers inside the brackets indicate the model speed up in comparison to the corresponding sequential simulation on a single CPU node).

4.1.5 *Hokkaido Nansei-Oki tsunami*

The magnitude M_s = 7.8 Hokkaido Nansei-Oki earthquake hit the west coast of Hokkaido, Japan on July 12 1993 and the small offshore island of Okoshiri in the Japan Sea, generating a severe tsunami. The disastrous event is simulated by the current tsunami model to test its performance for field-scale applications. Also the spurious waves generated at the open boundaries with abrupt changes are visualised, and the selected boundary treatment from Hou *et al.* (2015) is implemented to avoid the issue. The Disaster Control Research Centre (DCRC), Japan, digitised the bathymetric and topographic data which are available from the National Oceanic and Atmospheric Administration (NOAA) of the USA. DCRC also reconstructed initial wave profile with 4.9 m and 2.2 m uplift and 1.1 m subsidence (depression). Figure 4.17 shows the study area and the initial water level. The bathymetry with a 450 m resolution (379 × 687 cells) is adopted for the tsunami wave propagation simulation. Constant Manning coefficient of 0.015 $s/m^{1/3}$ is applied across the whole domain.

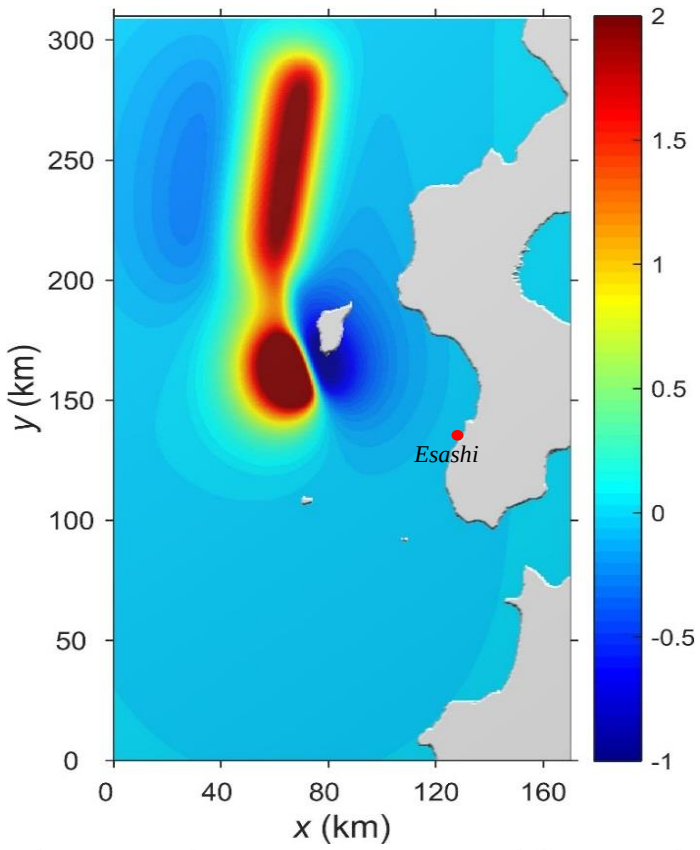
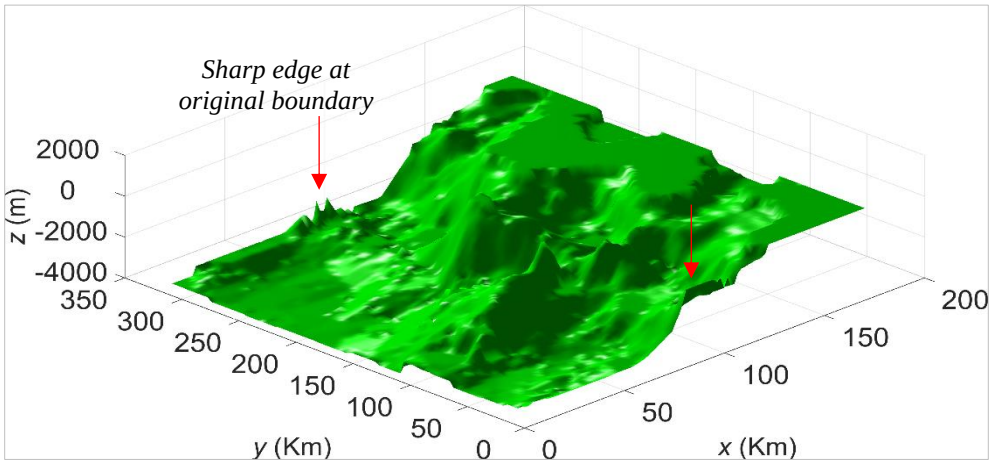


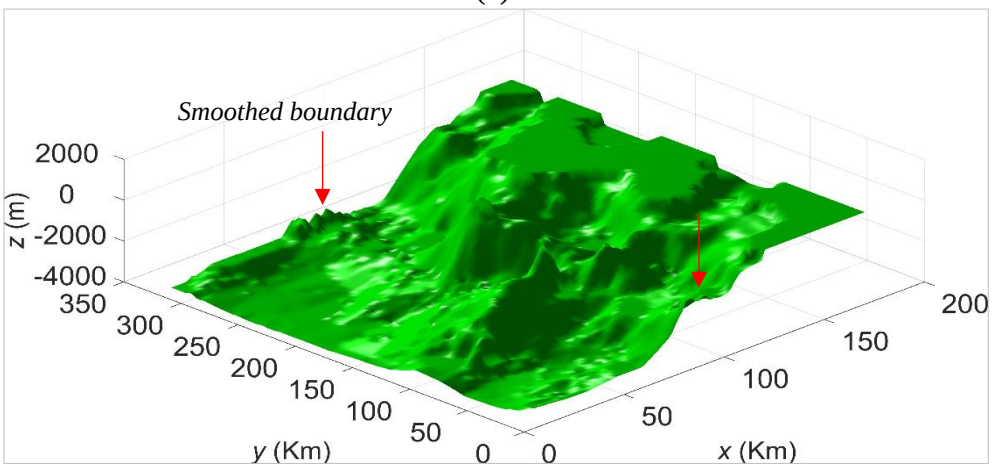
Figure 4.17 Hokkaido Nansei-Oki tsunami: computational domain and initial water level.

Figure 4.18 (a) shows the bathymetry/topography with original bed topography including sharp bumps in the northern and southern boundaries, while Figure 4.18 (b) represents the same bathymetry/topography with 10 extended cells/layers to smooth the boundary topography. Figure 4.19 shows the simulated results for wave propagation using the original boundary (Figure 4.19 (a), (b)) and the modified domain with extended cells/layers at the boundary

(Figure 4.19 (c), (d)) at t = 300 s and t = 500 s, respectively. For the simulation with the original boundary topography, as it is evident from Figure 4.19 (a), the propagated wave has reached the northern boundary and an unphysical wave is generated after 300 s due to the existence of the bumps at the north boundary. This unphysical wave then propagates back to the domain. After about 500 s, another spurious wave also appears at the south boundary, as shown in Figure 4.19 (b). After extending 10 cells/layers to the boundaries, the spurious waves are eliminated as presented in Figure 4.19 (c&d) and the numerical results become stable and satisfactory.



(a)



(b)

Figure 4.18 Hokkaido Nansei-Oki tsunami: (a) 3D view of original bathymetry/topography (b) 3D view of bathymetry/topography with extended cells/layers.

The 3D view of the tsunami wave propagation is illustrated in Figure 4.20, starting from the earthquake elapsed time up to 40 minutes, which are in good agreement with those reported in Hou *et al.* (2015). Figure 4.21 presents the numerically predicted history of water surface level at Esashi tide gauge, in comparison with the available measurements. From the figure, although the second-order accurate model can satisfactorily predict the wave peaks, there are evident discrepancies between measured and predicted wave arrival times. These discrepancies could be due to the difference of the exact locations of the gauges in the simulation compared to

reality, and also the limitations of the SWEs for near-coastal simulations. Moreover, in the field scale tsunami simulations most uncertainties arise from tsunami initializations. The potential source of errors for real case tsunami simulations are discussed in Chapter 5. The first-order accurate model presents strong numerical dispersion for this test and the result is basically not acceptable. Therefore, high-order accurate model, at least second-order, is recommended for large-scale tsunami simulations.

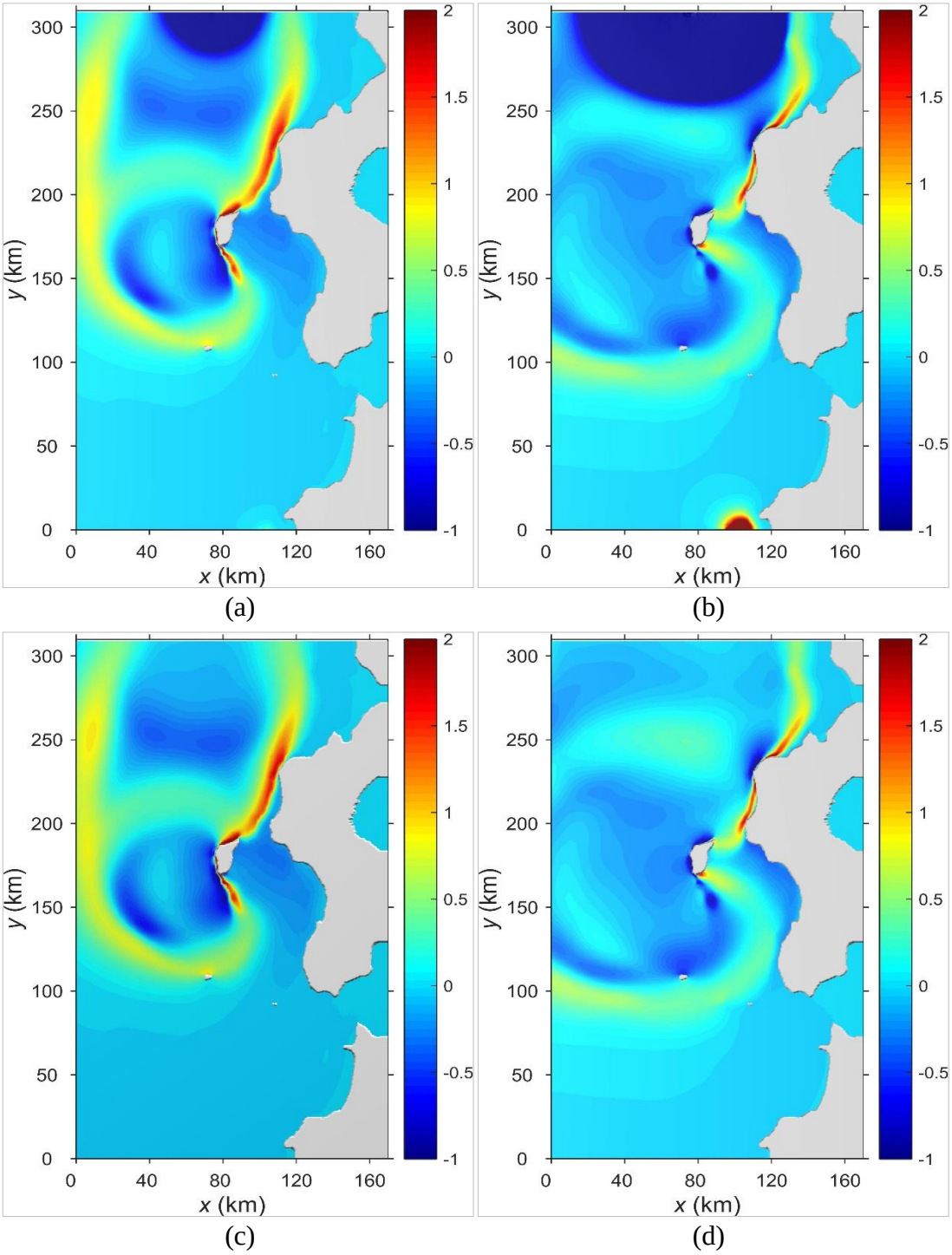
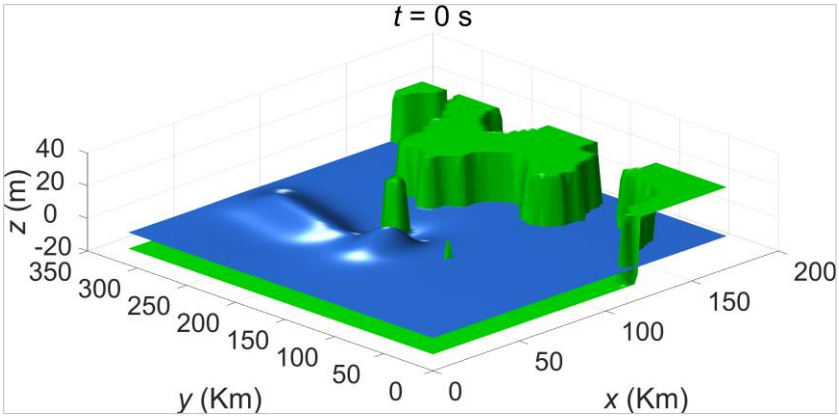
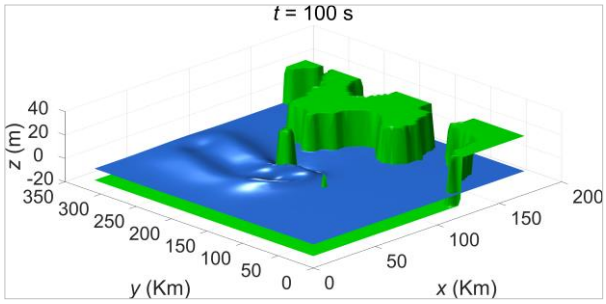


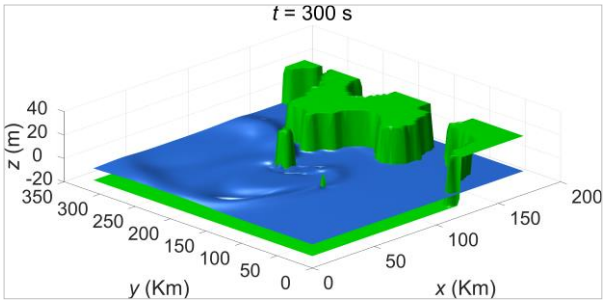
Figure 4.19 Hokkaido Nansei-Oki tsunami: 2D view of wave propagation at (a) t = 300 s with original boundary; (b) t = 500 s with original boundary; (c) t = 300 s with extended cells/layers at boundaries; (d) t = 500 s with extended layers at boundaries.



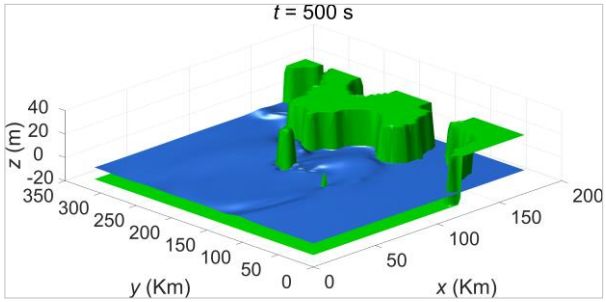
(a)



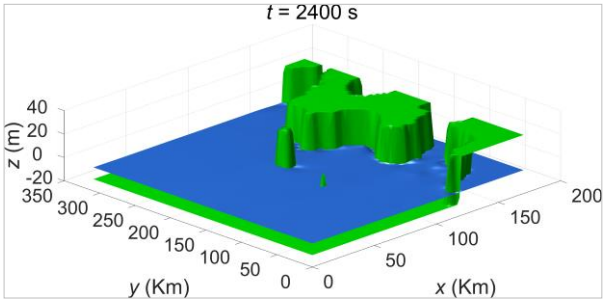
(b)



(c)



(d)



(e)

Figure 4.20 Hokkaido Nansei-Oki tsunami: 3D view of tsunami wave propagation at time (a) t = 0 s; (b) t = 100 s; (c) t = 300 s; (d) t = 500 s, (e) t = 2400 s.

Table 4.11 compares runtimes required by the 1st and 2nd-order accurate models to complete a 40-minutes simulation on different devices. Comparing with that on a single CPU core, the simulations with the second-order numerical scheme on four CPU cores, Geforce GTX 560 Ti GPU and the Tesla M2075 GPU are sped up by 3, 39 and 45 times, respectively, which is generally consistent with those reported in the previous benchmark tests. The high-performance first-order simulations present similar performance.

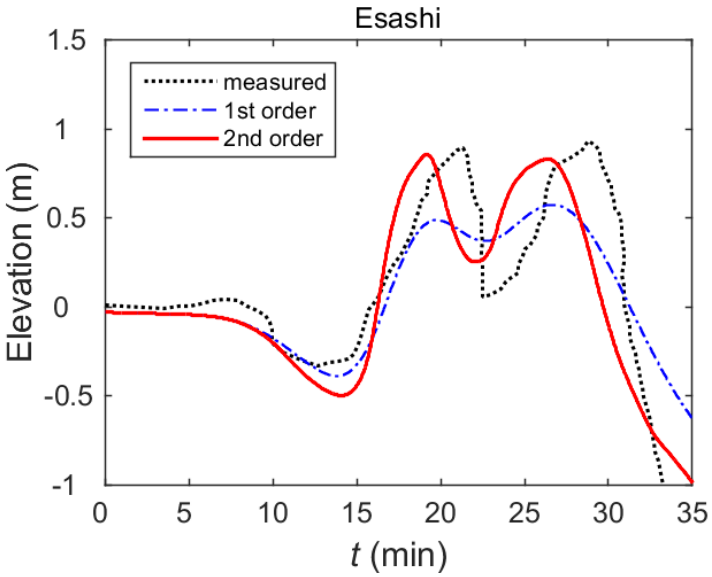


Figure 4.21 Hokkaido Nansei-Oki tsunami: time histories of water surface level predicted with the first and second-order accurate models, compared the measurements at Esashi tide gauge.

Order of accuracy	CPU	CPU+OpenMP	GPU GTX	GPU Tesla
1st	432.05 s (~1×)	140.14 s (~3.08×)	11.69 s (~36.96×)	9 s (~48×)
2nd	567.37 s (~1×)	181.25 s (~3.13×)	14.56 s (~38.97×)	12.56 s (~45.17×)

Table 4.11 Hokkaido Nansei-Oki tsunami: computation time on different hardware devices and the numbers inside the brackets indicate the model speed up in comparison to the corresponding sequential simulation on a single CPU node.

4.2 Summary

To validate different implementations of the current tsunami model, five test cases are simulated, including the idealized circular dam break, dam break over complex topography, runup of a solitary wave over a conical island, laboratory tsunami benchmark test and a field-scale tsunami event. The current GPU based hydrodynamic model based on the 2nd-order accurate scheme is capable of simulating tsunami propagation and run-up involving advancing bores and moving shorelines in the inundation zone over irregular topographies. For all test cases, satisfactory numerical results are obtained, compared with the laboratory measurements or alternative numerical predictions.

Compared with the single-core sequential simulation, four-core CPU parallel computing achieved via OpenMP improves the computational efficiency by more than 3 times, as it is expected. The model accelerated through heterogeneous parallel computing using GPU save more than 40 times of computational time, comparing with the single-core sequential simulation. The performance of the GPU simulations seems to increase with more computational cells used for the calculations, as the heavier use of GPU time balances the CPU-GPU communication

overhead which was has also noted by other researchers e.g. (Brodtkorb *et al.*, 2010; Vacondio *et al.*, 2014). Concerning the two different types of GPUs, the Tesla M2075 GPU improves the performance of the model slightly compared with the much cheaper desktop Geforce GTX 560 Ti GPU. However, because they are commonly installed with more advanced architectural design containing much larger memory, the professional GPUs are more suitable for large-scale high-resolution practical simulations for assessing more reliably the impact of tsunami waves on people, urban buildings and infrastructure systems.

The numerical simulations of wave amplitude and arrival time are sensitive to the change of spatial resolution and the numerical order of accuracy. The 2^{nd}-order accuracy is better in resolving the sharp-fronted flow hydrodynamics with less numerical dissipation. The importance of the 2^{nd}-order accuracy was remarkable for the field-scale simulation of Hokkaido Nansei-oki tsunami, which will be also noticed for the 2011 Japan tsunami in next chapter. This essentially indicates that results of similar accuracy may be theoretically obtained using a high-order scheme at relatively lower spatial resolutions. Considering tsunami waves are typically characterized as wall of water rapidly moving inland over a vast domain, higher, at least a 2nd-order numerical scheme may be essential for reliable predictions.

Chapter 5 Case Study of Tohoku Japan Tsunami, 2011

The real-world tsunamis generated in the deep ocean can travel with a high speed in the open water and reach the coastal regions in a short time with a large wave magnitude due to the shallow water effects, causing flood devastation. Therefore, fast estimation of tsunami wave propagation, including the wave amplitude and arrival time, is vital for evacuation and hazard mitigation purposes. The aim of this chapter is to demonstrate the performance of the GPU-based tsunami model introduced in the previous chapter in terms of accuracy and computational efficiency for reproducing a large-scale transoceanic tsunami event that hit Eastern Japan in 2011. The focus is mainly on simulating the propagation of the tsunami wave to further validate the model by comparing the results with the field measurements or alternative numerical predictions available in the literature. Two different fault models for tsunami initialization are introduced and reviewed; the Okada model is then adopted to estimate the vertical displacement of water surface and provide the initial conditions to drive the tsunami simulations. Sensitivity analysis is carried out for different spatial resolutions and slope limiters, aiming to find out the optimised resolution and limiter for accurate simulations. Finally the much improved computational efficiency of the GPU-based tsunami model is confirmed by comparing the required runtimes between the current model and its counterpart running on a single CPU. Part of results presented here can be also found in (Amouzgar *et al.*, 2016).

5.1 Introduction

At 14:46:18 local time on March 11, 2011 a mega earthquake ($M_w = 9.0$) occurred off the Pacific coast of Tohoku, Japan. The epicentre was at (38.1035° N, 142.861° E) according to Japan Meteorological Agency (JMA), as shown in Figure 5.1. This generated a huge tsunami and caused over 15,000 casualties and 220 billion US dollars of damage. This disaster showed the vulnerability of coastal areas to natural hazards such as tsunamis, even in well prepared countries as Japan. Due to the extensive measurements and data from different sources, this disastrous event is the best ever documented large tsunami and earthquake in history. The rich sources of datasets from this event include tsunami wave measurements, flood extent, run-up records, damages, ground deformation, etc., enabling researchers and scientific communities to better understand this and other similar extreme events and enhance their capability to mitigate the consequences. In this regard, much research has been reported to date for different aspects of this event and intensive research is still ongoing, e.g. (Fujii *et al.*, 2011; Lay *et al.*, 2011a; Maeda *et al.*, 2011; Yamazaki *et al.*, 2011b; Chan and Liu, 2012; Grilli *et al.*, 2012; Popinet,

2012; Hooper *et al.*, 2013; Melgar and Bock, 2013; Satake *et al.*, 2013; Wei *et al.*, 2013; Yamazaki *et al.*, 2013; Goda *et al.*, 2014; Tappin *et al.*, 2014; Wei *et al.*, 2014).

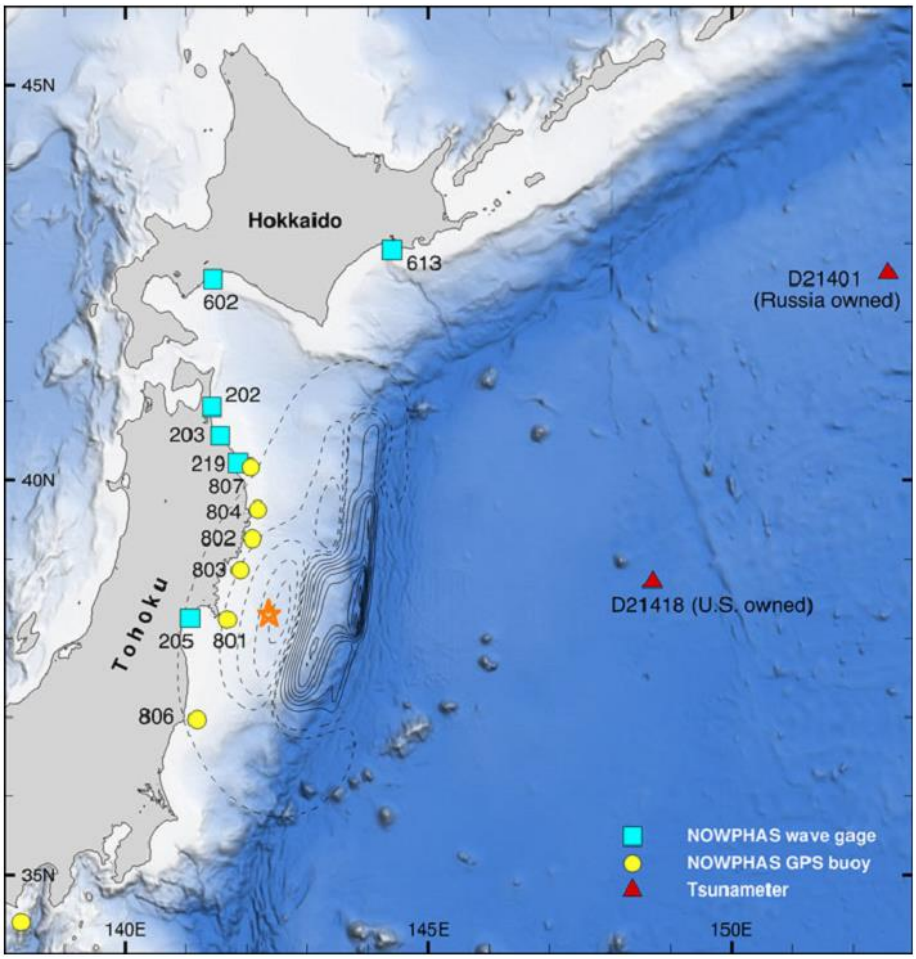


Figure 5.1 Epicentre of the 11 March 2011 Japan earthquake (orange star) reported by USGS. Also shown are the locations of deep-ocean tsunameters, nearshore GPS buoys, and wave gauges (Wei *et al.*, 2013).

5.2 Research area

The study area is fit into a 1350 km × 1822.50 km rectangular domain of a Cartesian coordinate system, as shown in Figure 5.2. Measurements in terms of tsunami waveform are available for the current study from four types of gauges in 15 locations from nearshore to offshore. These gauges include six wave gauges close to the coast, six GPS buoys at nearshore which are provided by Japan's Nationwide Ocean Wave information network for ports and HArbourS (NOWPHAS), two cabled pressure gauges, and one DART buoy (Deep-ocean Assessment and Reporting of Tsunamis) that is 500 km away from the epicentre. Details of these gauges are provided in Table 5.1 and illustrated in Figure 5.1 (Geographic coordinate system) or in Figure 5.2 (Cartesian coordinate system).

The bathymetry/topography and other necessary data for this study are provided by the Disaster Prevention Research Institute (DPRI) in Kyoto University, Japan. The provided resolution of bathymetry/topography data is 1,350 m for the whole domain and 450 m at a regional scale, other bathymetry resolutions are computed by bilinear resampling method with ArcGIS tool. A constant Manning coefficient of 0.025 $m/s^{1/3}$ is used across the entire domain.

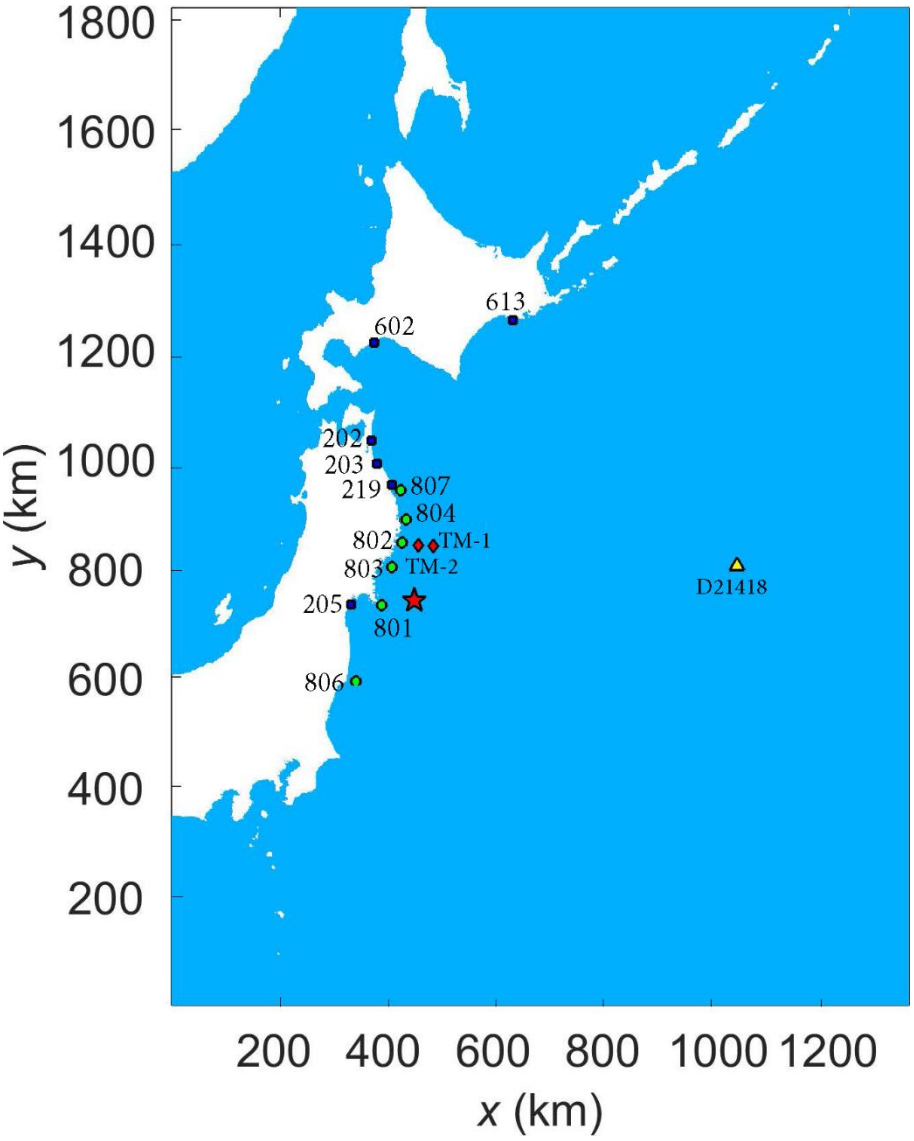


Figure 5.2: Computational domain for the 2011 Japan tsunami and locations of gauges where the green filled circles represent the GPS buoys, dark blue squares the wave gauges, red filled diamonds the pressure gauges, and the yellow filled triangle the deep-ocean tsunameter. The red star represents the epicentre reported by USGS.

5.3 Tsunami generation and model initialization

Tsunamis can be generated by disturbances above or below the water and this study considers the tsunamis generated by earthquakes. The earthquake slip is responsible for rapid seafloor deformation which should be determined for accurate tsunami forecasting. In this regard, there are various earthquake slip models in the literature for estimating the slip distribution, which

are commonly based on the tsunami waveform data (Fujii *et al.*, 2011; Satake *et al.*, 2013), teleseismic data (Hayes, 2011; Shao *et al.*, 2011), recorded onland deformation by GPS Earth Observation Network (GEONET) (Simons *et al.*, 2011; Iinuma *et al.*, 2012) or the combination of these observations for joint inversions.

Gauge name	Type	Depth (m)	Lat. (°N)	Lon. (°E)	*x* (m)	*y* (m)
807-Iwate North	GPS buoys	125	40.11667	142.06667	420,460	941,250
804-Iwate Central	GPS buoys	200	39.62722	142.18667	430,190	887,060
802-Iwate South	GPS buoys	204	39.25861	142.09694	422,080	846,050
803-Miyagi North	GPS buoys	160	38.85778	141.89444	404,070	801,370
801-Miyagi Central	GPS buoys	144	38.2325	141.68361	384,780	731,830
806-Fukushima	GPS buoys	137	36.97139	141.18556	338,490	591,710
613-Kushiro	wave gauge	50	42.8844	144.410	625,170	1,251,200
602-Tomakomai	wave gauge	50.7	42.5427	141.441	371,990	1,210,100
202-Mutsu Ogawara	wave gauge	43.8	40.9347	141.415	366,560	1,031,600
203	wave gauge	27.7	40.5528	141.546	376,890	989,260
219	wave gauge	49.5	40.2111	141.862	403,160	951,540
205	wave gauge	21.3	38.2412	141.047	329,090	732,580
TM1	Pressure gauge	1600	39.1855	142.767	479,870	838,850
TM2	Pressure gauge	1000	39.2036	142.444	451,990	840,360
D21418	DART	5660	38.821	148.655	1,034,135	803,672

Table 5.1: Details of the gauges used in this study for the 2011 Tohoku tsunami simulation.

In this research, the tsunami source/initialisation is determined from the results of a tsunami waveform inversion approach reported by Fujii *et al.* (2011) and from the tsunameter-derived tsunami measurements in the deep ocean provided by Wei *et al.* (2014).

The initial vertical sea level displacement is assumed to be equal to the total vertical seafloor deformation estimated by the source/fault model. According to this assumption, the initial water surface displacement to initiate a tsunami is calculated using the Okada rectangular fault model (Okada, 1985) in this work. The Okada model is developed based on the assumption of a rectangular fault plane buried in an elastic semi-infinite half space. The Okinv3 code provided by Clarke *et al.* (1997) is used to compute the initial condition for the 2011 Japan tsunami in this work.

A typical rectangular fault is shown in Figure 5.3 to illustrate the fault geometry and parameters. The colliding interface between subducting and overriding plates defines the fault plane (Figure 5.3 (a)). To implement the fault model in Okinv3, nine parameters for each subfault must be known including strike angle θ, dip angle δ, rake angle λ, slip dislocation (m), longitude and latitude of the projected centre (degrees), length of scarp L (km), minimum and maximum depth of dislocation (km). The fault angles are shown in Figure 5.3 (b). Strike is the angle measured clockwise from the North to strike direction ($0^o \leq \theta \leq 360^o$). Dip is the angle that defines the steepness of the fault surface ($0^o \leq \delta \leq 90^o$). Slip angle (rake angle) is the angle describing the motion of the material above the fault plane relative to that from the below, measured anti-clockwise in the fault plane from the strike direction ($-180^o \leq \lambda \leq 180^o$). Slip dislocation is the distance of relative motion which is measured on fault plane. Figure 5.3 (c) illustrates the projection of the fault plane on the earth surface, where 'h' is the focal depth the vertical distance between the centre of the fault plane and the epicentre.

Following the definition of the parameters for all subfaults, Okinv3 will compute the final deformation based on the Okada model in a short time so the vertical displacements can be used as the initial conditions in the GPU-based tsunami model.

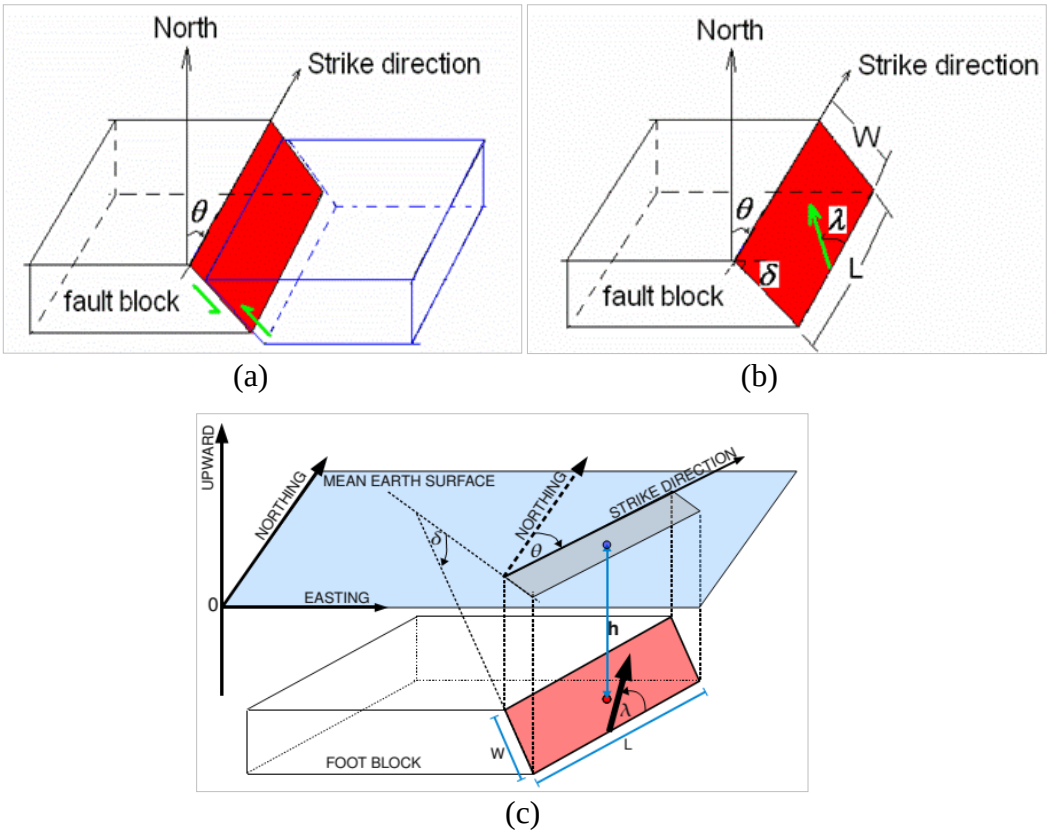


Figure 5.3: Okada model: (a) fault plane as shown in red; (b) fault geometry (c) projection of fault on earth surface (Wang, 2009).

5.3.1 *Fault parameters by tsunami waveform inversion*

Fujii *et al.* (2011) provided the slip distribution estimated using tsunami waveform inversion from the available gauges, showing a large slip of over 40 m along the trench axis. The tsunami source is divided into 40 subfaults (50 km × 50 km) to cover the aftershock area as shown in Figure 5.4 and Table 5.2.

The top depth is assumed to be 0 km, 12.1 km, 24.2 km and 36.3 km for the near-trench, shallow, middle and deep sub-faults, respectively. The USGS W-phase moment tensor solution estimated the fault angles as: strike 193°, dip 14°, slip 81°.

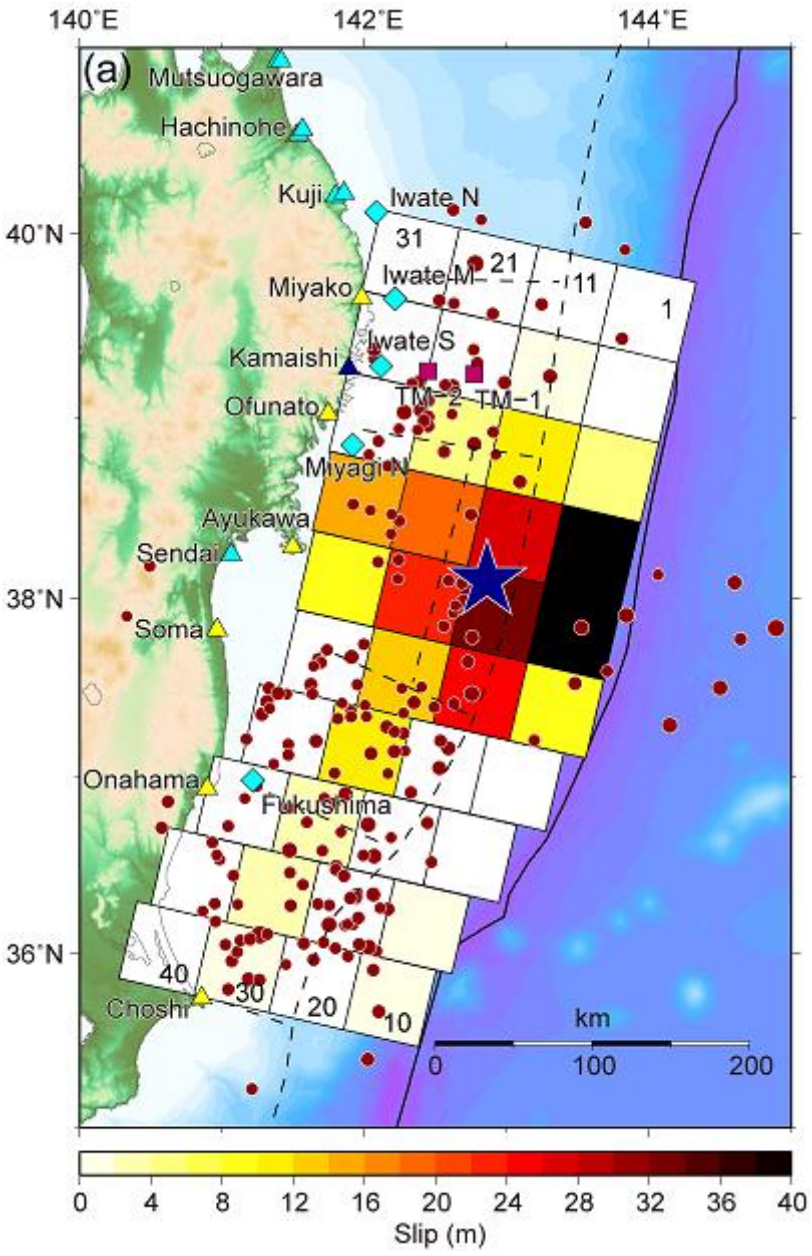


Figure 5.4: Slip distributions predicted by tsunami waveform inversion by (Fujii *et al.*, 2011). The order of numbering of the 40 subfaults can be inferred from the numbered subfaults in the northernmost and southernmost.

Based on this information, the instantaneous vertical displacement of the seafloor is computed and shown in Figure 5.5. According to the computation, a maximum uplift of about 12 m near to the trench and a depression of about 2 m in the down-dip region is estimated.

No.	Lat. (°N)	Lon. (°E)	Depth (Km)	Slip±error (m)
1	39.738	144.331	0	0.00±0.23
2	39.300	144.20	0	0.00±0.15
3	38.862	144.069	0	5.66±4.85
4	38.424	143.939	0	41.15±17.14
5	37.986	143.810	0	47.93±21.49
6	37.547	143.682	0	8.44±4.25
7	37.135	143.400	0	0.00±1.81
8	36.730	143.070	0	0.00±0.87
9	36.325	142.740	0	1.89±0.75
10	35.905	142.504	0	0.63±0.97
11	39.836	143.778	12.1	0.00±0.03
12	39.398	143.651	12.1	0.81±0.44
13	38.960	143.523	12.1	10.67±6.16
14	38.522	143.397	12.1	27.84±11.09
15	38.084	143.271	12.1	33.79±18.47
16	37.646	143.146	12.1	24.11±9.30
17	37.233	142.867	12.1	0.00±2.5
18	36.828	142.540	12.1	0.00±0.32
19	36.423	142.213	12.1	0.00±0.24
20	36.003	141.979	12.1	0.00±0.91
21	39.934	143.224	24.2	0.00±0.41
22	39.496	143.100	24.2	0.00±0.27
23	39.058	142.977	24.2	4.86±3.99
24	38.620	142.853	24.2	19.56±9.68
25	38.182	142.731	24.2	23.38±12.54
26	37.744	142.609	24.2	13.13±5.81
27	37.331	142.333	24.2	11.13±5.44
28	36.926	142.009	24.2	2.23±1.66
29	36.521	141.684	24.2	2.25±2.02
30	36.101	141.454	24.2	0.54±0.31
31	40.032	142.670	36.3	0.00±0.42
32	39.594	142.549	36.3	0.00±0.17
33	39.156	142.430	36.3	0.00±0.55
34	38.718	142.309	36.3	14.64±8.98
35	38.280	142.190	36.3	9.46±4.44
36	37.842	142.071	36.3	0.00±0.00
37	37.429	141.798	36.3	0.00±1.88
38	37.024	141.477	36.3	0.00±0.07
39	36.619	141.155	36.3	0.00±0.00
40	36.199	140.928	36.3	0.00±1.22

Table 5.2: Slip distribution estimated by tsunami waveform inversion (Fujii *et al.*, 2011). The latitude and longitude as provided represent the northeast corner of each subfault.

For simplicity, the results reproduced from the fault information and parameters based on tsunami waveform inversion provided by Fujii *et al.* (2011) will be referred to as 'source 1' from now on.

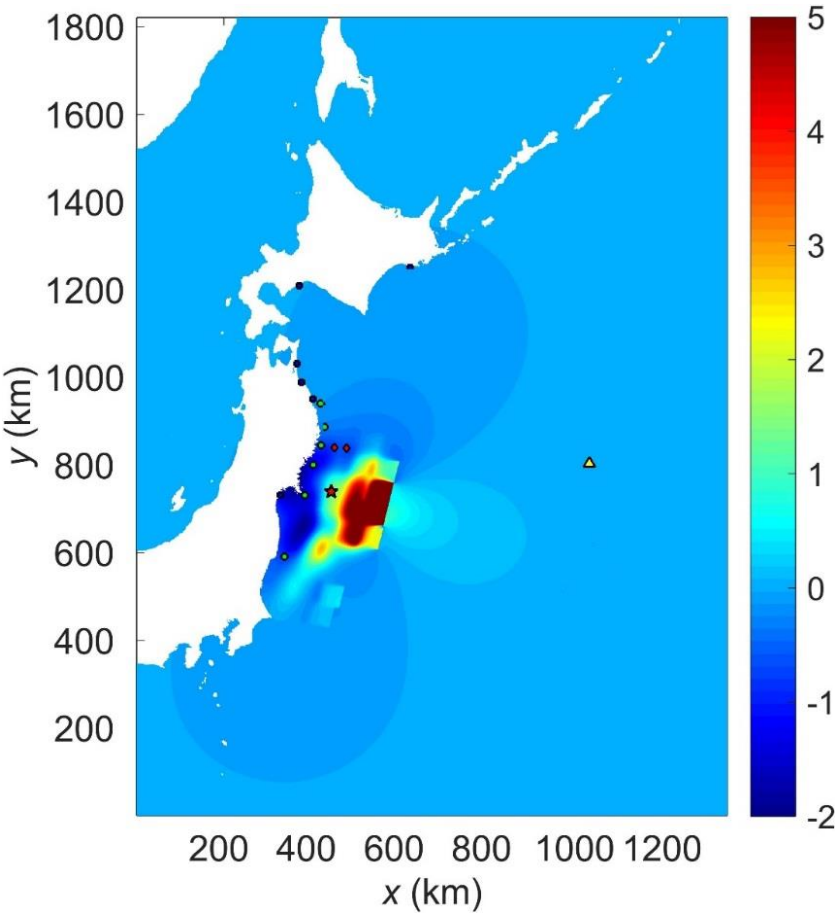


Figure 5.5: Instantaneous seafloor (and hence water level) deformation of the 2011 Tohoku earthquake computed by Okada's solution ('source 1'). The fault parameters and information are estimated by tsunami waveform inversion as provided by Fujii *et al.* (2011). The red star indicates the epicentre provided by the USGS.

5.3.2 *Fault parameters from tsunameter measurements*

In this section, the tsunameter inversion method for deriving the initial tsunami conditions is briefly described. The information from two deep ocean tsunameters D21418 and D21419 is used by the NOAA Centre for Tsunami Research (NCTR) for a real time inversion based on the model developed by Percival *et al.* (2011), as shown in Figure 5.6. This method is dependent on the available database of precomputed tsunamis from subfault patches parametrised to produce global subduction zones (Gica *et al.*, 2008). The dimensions of the subfaults are 100 × 50 km; for the 1 m slip, each unit source represents a 7.5 magnitude earthquake. The patches from eastern Japan to the trench and plate boundaries are illustrated in Figure 5.6. The tsunami wave propagation for these unit sources is predicted by the MOST tsunami model and recorded in the tsunami propagation data base. The tsunami source is then estimated by the precomputed

tsunamis which are used together with the real time tsunami records from bottom pressure gauges in the deep ocean (Tang *et al.*, 2009; Titov, 2009). During the 2011 Tohoku earthquake the first preliminary inversion was made after 56 min of elapsed time from station D21418, 500 km eastward of the epicenter. Then the second inversion was carried out 90 minutes after the earthquake, when the first wave passed station D21401 that is 990 km northeast of the epicenter. The final results of the model presented the deformation of the seafloor best fitted by a source rupture 400 km along strike and 100 km toward Japan from the trench, as shown in Figure 5.6. The information from this method was used by a number of other researchers to study the 2011 Japan tsunami. For example Wei *et al.* (2013) and Wei *et al.* (2014) used this tsunami source to provide the initial conditions for the MOST tsunami model. The refined estimates of the fault parameters are provided in Table 5.3. With the information as provided, the sea level variation is computed by the Okada model and visualized in Figure 5.7. The results show a maximum uplift of about 10 m near to the trench and a depression of about 2 m in the down-dip region. For simplicity, the results reproduced from the fault information and parameters based on tsunameter measurements provided by Wei *et al.* (2013) are called 'source 2' from now on.

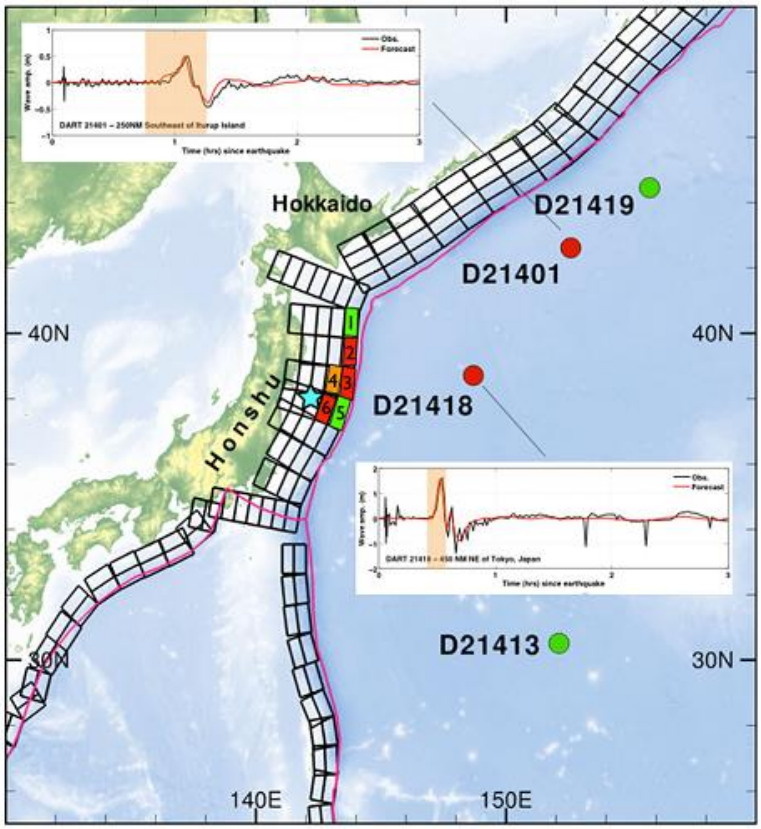


Figure 5.6: Tsunami source from tsunameter measurements for the 2011Japan tsunami. The orange area in the plots for stations D21418 and D21401 shows the interval of the time series used in the inversion. The star indicates the epicentre reported by USGS. The purple lines are the plate boundaries. The black boxes represent the tsunami unit sources precomputed in NCTR'S database. The numbers in boxes indicate the sources number in Table 5.3 (Wei *et al.*, 2014).

source	Lat. (°N)	Lon. (°E)	Strike	Dip	Rake	Depth (km)	Slip (m)
1	40.3125	143.5273	185	19	90	5.0	4.66
2	39.4176	143.4246	185	19	90	5.0	12.23
3	38.5254	143.2930	188	19	90	5.0	21.27
4	38.5837	142.7622	188	21	90	21.3	26.31
5	37.6534	143.0357	198	19	90	5.0	4.98
6	37.7830	142.5320	198	21	90	21.3	22.75

Table 5.3: Tsunami source constrained from the two tsunameters D21418 and D21401 in the deep ocean (Wei *et al.*, 2014). Each subfault is 100 km long and 50 km wide. The indexes in the first column correspond to the numbers in the boxes as shown in Figure 5.6.

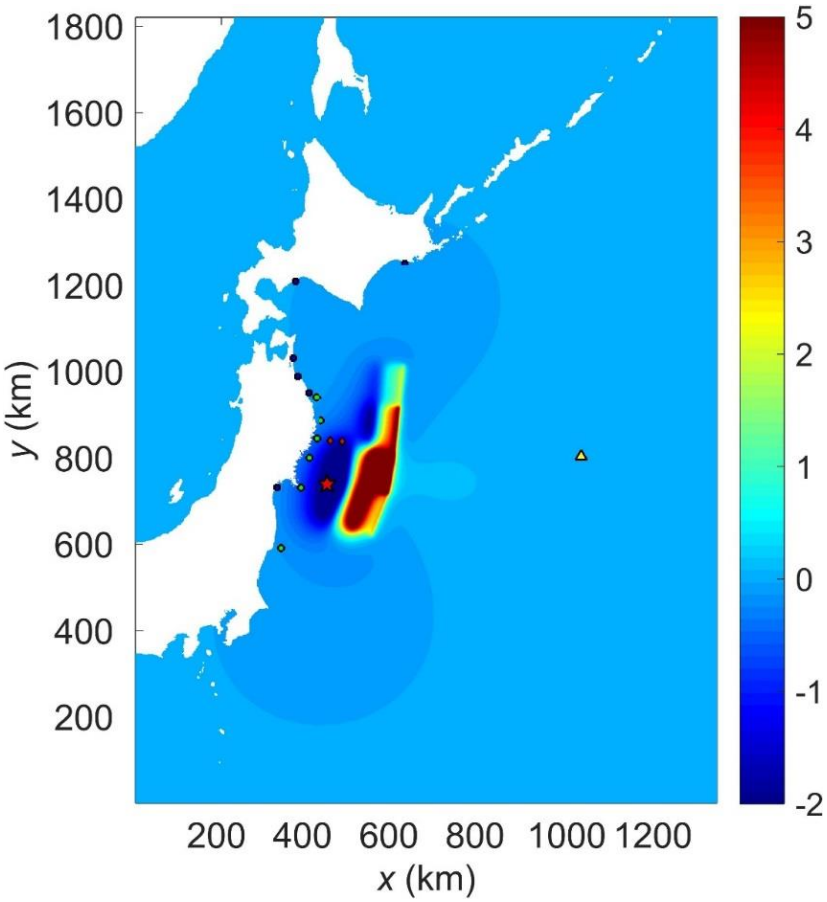


Figure 5.7: Instantaneous seafloor (and hence water level) deformation of the 2011 Tohoku earthquake computed by the Okada model ('source 2'). The tsunami source inferred from measurements of tsunameter measurements in deep-ocean is provided by Wei *et al.* (2014).

5.4 Simulation of tsunami wave propagation

In this section, the 2011 Japan tsunami is reproduced based on the two aforementioned source models. To verify the numerical results, the time series of wave propagation at different gauges are compared with the measurements available in a number of gauges. Sensitivity tests are then carried out to investigate the effects of spatial resolution and the choice of different numerical limiters.

5.4.1 *2D Wave evolution*

The tsunami wave propagation at the first 80 minutes based on the initial conditions provided by 'source 1' and 'source 2' is respectively presented in Figure 5.8 (Page 104) and Figure 5.9 (Page 105). The wave evolution and the associated arrival time are consistent for both simulations. After being initiated by the earthquake, the tsunami wave propagates radially toward the east coast of Japan and into the deep ocean in the other direction. In less than half an hour, the first leading high wave reaches the coast, consistent with the records at the wave gauges. The radial and long-crested waves reach the Iwate-Central (804) and Iwate-south (802) gauges after about t = 20 min and the long-crested wave propagates toward the Iwate-North (807) gauge (refer to Figure 5.2 and Table 5.1 for the locations). In about t = 30 min, the wave has reached Miyagi-Central (801) and the Fukushima (806) stations toward the south while by this time it has also reached the deep ocean tsunameter D21418 gauge located 500 km east of the epicenter. By t = 80 min the wave has covered most of the east coast of Japan followed by the reflected and diffracted waves which caused the second flood wave on the Sendai coast close to Miyagi central gauge. The open boundaries are adopted for this real-world case, and with regard to the open boundary treatment, the simulated results also show that the propagating waves pass through the open boundaries of the computational domain successfully without inducing any numerical instability. In this real-world test it is found, after checking various spatial resolutions in the range of 450 m to 3.6 km, that 20 cells are required to extend and smoothen the bed topography at the boundaries to prevent spurious waves.

The current numerical prediction is found to be similar to the animation of the wave evolution provided in the electronic supplementary material of Fujii *et al.* (2011) and Wei *et al.* (2013). Direct comparison can be also made with those results presented in Yamazaki *et al.* (2011b), again showing good agreement.

5.4.2 *Model-observation comparison*

Figure 5.10-5.13 (Pages 109-110) present the simulation results in terms of time histories of water surface oscillation at the 15 gauges (4 different types), compared with the measurements. The predicted and measured amplitudes of the leading wave and the corresponding arrival times, as well as the relative errors, are respectively shown in Table 5.4 (Page 110) and Table 5.5 (Page 111) for the simulations based on 'source 1' and 'source 2'. In order to make a consistent comparison, all of the simulations in this section are undertaken for both cases using the same grid resolution (1350 m) and using the minmod slope limiter.

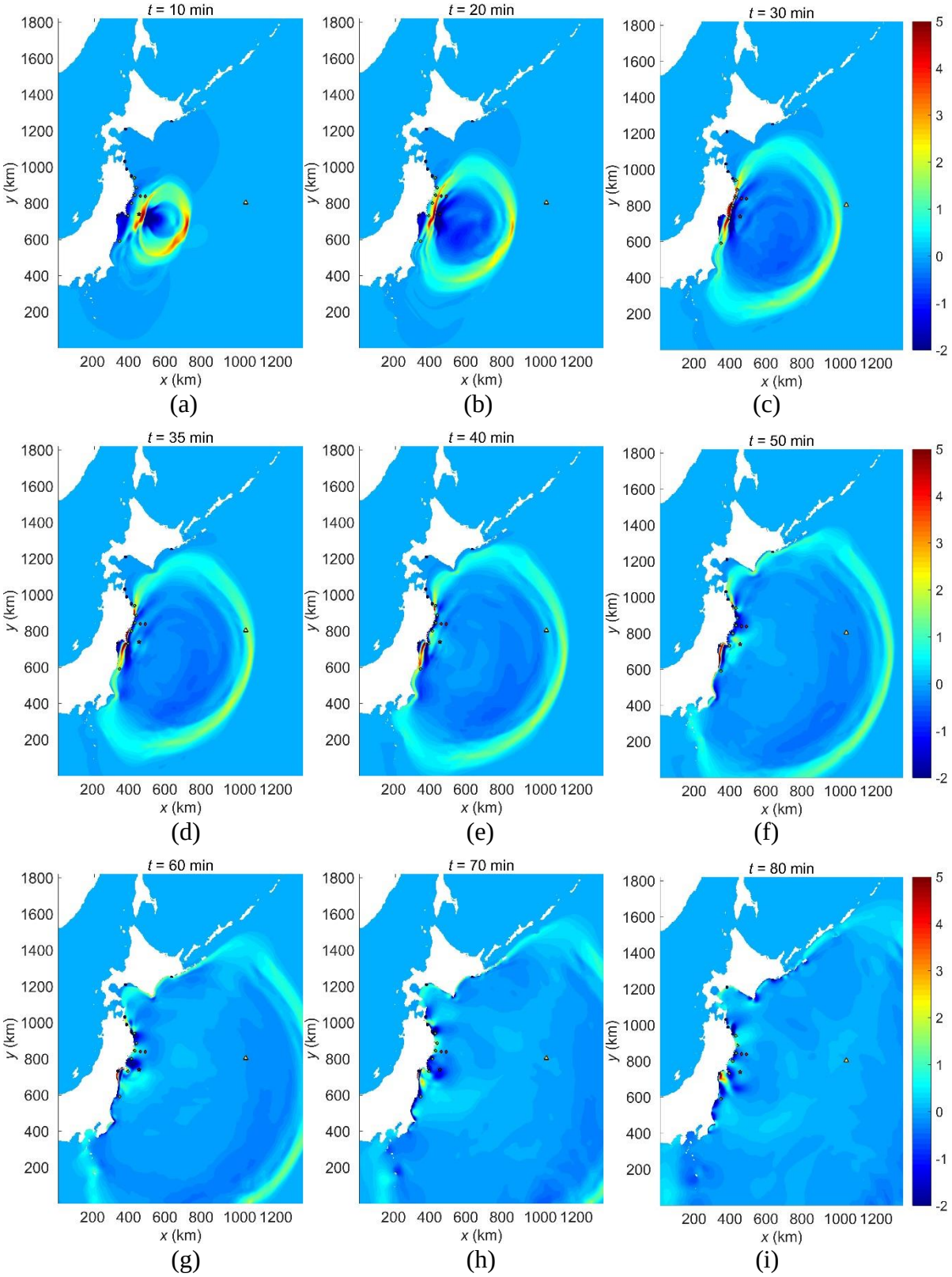


Figure 5.8: Evolution of tsunami wave based on the initial conditions provided by 'source 1' at different times: (a) t = 10 min; (b) t = 20 min; (c) t = 30 min; (d) t = 35 min; (e) t = 40 min; (f) t = 50 min; (g) t = 60 min; (h) t = 70 min; (i) t = 80 min.

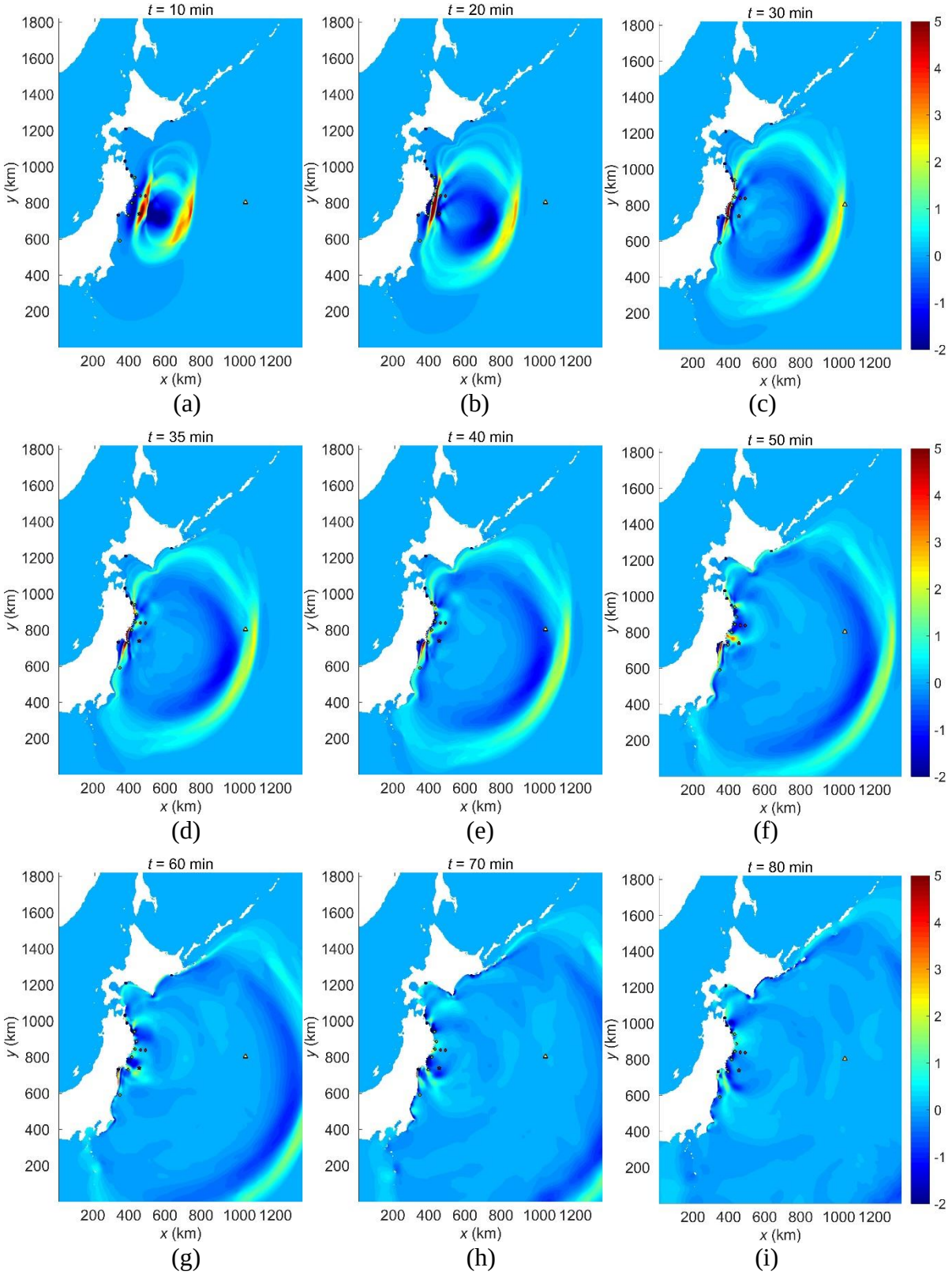


Figure 5.9: Evolution of tsunami wave based on the initial conditions provided by 'source 2' at different times: (a) t = 10 min; (b) t = 20 min; (c) t = 30 min; (d) t = 35 min; (e) t = 40 min; (f) t = 50 min; (g) t = 60 min; (h) t = 70 min; (i) t = 80 min.

The model-observation comparison are categorized for 6 GPS buoys, 6 coast gauges, 2 bottom pressure gauges, and 1 DART station as following:

- GPS buoys (807, 804, 802, 803, 801, 806)

Because all GPS buoys are deployed at water depth ranging from 100 to 200 m, their measurements are not influenced by the coastline. Therefore the measurements at these buoys may be considered to represent the direct information about the source of the tsunami.

As shown in Figure 5.10 (Page 109), both source models can produce reasonable estimates of the wave amplitude and period including the trailing waves compared with the observation. From north to south, the waves at gauges 807, 804, 802 in the Iwate region are underestimated about 35% and 14% on average by the simulations based on 'source 1' and 'source 2' respectively. 'Source 1' predicts the arrival times that agree slightly closer to the observations, compared with 'source 2'.

The first leading wave that was recorded to be over 6 m at 803 and 801 in the Miyagi region is also captured by both of the 'source 1' and 'source 2' simulations properly. The high wave amplitudes at these locations are consistent with high run-up height and inundation depth recorded along the coast between latitude 38° N and 40° N (Mori *et al.*, 2011). The wave amplitude decays toward the south to 2 m as shown for station 806 in the Fukushima region. The wave peak estimated by the simulation based on 'source 1' is similar to the observations with a slight phase difference while the simulation based on 'source 2' underestimate the wave peak by about 40%. These simulations are extended up to 6 hours for a better comparison with the observations and the numerical results show that the following trailing waves have also been reproduced favourably by the current tsunami model. The trailing waves are probably affected more by wave interactions such as wave refraction, reflection and shoaling with the local bathymetry and coastal features, rather than by the original deformation of sea surface that initiates the tsunami. So the use of different source models does not seem to have a major effect of the prediction of trailing waves.

- Near coast wave gauges (613, 602, 202, 203, 219, 205)

The near shore gauges which are located in the depth between 20 and 50 m between north of Tohoku and south of Hokkaido also recorded the tsunami waves. However, three of these gauges including 203, 219, and 205 stopped working shortly after the first tsunami wave reached the gauges as shown in Figure 5.11 (Page 109). Still the first recorded wave peak could

be compared with the simulation results. 'Source 1' can produce a similar height as observed while 'source 2' slightly underestimates the first peak for these three gauges.

The predicted results at the other wave gauges including 613, 602, and 602 which were fully operating during the tsunami event overall show good agreement with the measurements in terms of waveform, amplitude and phase for both models. The current numerical results are observed to be similar to those reported by Wei *et al.* (2013). It should be mentioned that the source of errors for accurately predicting tsunami wave dynamics at these near-shore gauges may be linked to the shallow bathymetry. The simulations in the shallow coastal regions are expected to involve larger uncertainties than those in the the deep water due to the use of relatively coarse resolution topograhy/bathymetry data in the simulations.

- Bottom pressure gauges (TM-1, TM-2)

The cable type ocean bottom pressure (TM-1 and TM-2) located at medium depth of more than 1000 m recorded two stages of tsunamis with a gradual sea level rise of 2 m followed by an impulsive tsunami wave with extra 3 m in amplitude as shown in Figure 5.12 (Page 110). While the waveform is successfully reproduced by the simulation based on 'source 1', the peak is underestimated about 30 %. The underestimation of wave peak at this gauges was also reported by Fujii *et al.* (2011) using the same source information but a different hydrodynamic model. On the other hand the source 'source 2' can reproduce wave amplitudes of over 4 m, which is in close agreement with the recorded tsunami wave.

- Deep ocean tsunameter (D21418)

As stated by Wei *et al.* (2014), the time series recorded by the tsunameters are not only used for deriving the tsunami source but can also provide data for validating the model for tsunami forecasting. As shown in Figure 5.13 (Page 110), the simulations based on both of the source models are able to predict the arrival time and the leading peak that are close to the observations. However, the results based on 'source 1' show a slightly underestimated peak and an arrival time that is 2 minutes later than observed. On the other hand, the simulation based on the 'source 2' predicts better results in comparing to the measurements.

The current numerical results discussed here also compare favourably with the model predictions presented by other researchers, e.g., (Fujii *et al.*, 2011; Yamazaki *et al.*, 2011b; Grilli *et al.*, 2012; Popinet, 2012; Satake *et al.*, 2013; Wei *et al.*, 2013; Goda *et al.*, 2014; Wei *et al.*, 2014).

5.4.3 *Evaluation of the source models*

The results from the proposed hydrodynamic tsunami model presented here show that on average both 'source 1' and 'source 2' can reproduce the tsunami wave sucessfully. Still the initial source model plays an important role in the simulation results and can be improved by ongoing research since the event happened. For example Satake *et al.* (2013) extended the tsunami source 'source 1' from 40 to 55 subfaults to estimate the slip distribution. Also a transient rupture time up to 5 minutes was assumed instead of instantaneous rupture on the fault. Therefore, by defining a multiple window analysis in the inversion, both temporal and spatial distributions of slip on subfaults become possible. Then the effect of the large displacement of the seafloor slope near the trench is considered which is reported to be responsible for 20%-40% of the tsunami amplitudes induced by the Tohoku earthquake (Satake *et al.*, 2013).

In order to evauate the results based on 'source 1' and demonstrate the sensitivity of the simulation results to the choice of source models, a simulation has also been carried out with initial tsunami conditions imposed by an alternative source model that is reinforced by tsunameter measurements ('source 2'), with an instantaneous rupture condition assumed for a consistent comparison. To evaluate the effects of the two tsunami sources quantitatively, the information of the leading wave including the maximum wave amplitude and the associated arrival time resulting from 'source 1' and 'source 2' are presented in Table 5.4 (Page 110) and Table 5.5 (Page 111) respectively. The relative error between predictions and measurements is calculated at each gauge by:

$$E = (A_m - A_{obs}) / A_{obs} \;;\; Accuracy = 1 - (\sum |E|) / n$$

where E is the relative error, subindices m and obs represent the 'model' and 'observation', A is either tsunami amplitude or the associated arrival time, '*Accuracy*' of the model represents the average error of the results obtained from the hydrodynamic model compared to the observation, and n is the number of stations used for the error calculation.

The quantitative values show that the prediction of tsunami arrival time are 'accurate' to over 90% for both 'source 1' and 'source 2'. The results also indicate that the model can predict the first wave amplitude with the average accuracy of 77 % and 83 % for 'source 1' and 'source 2' respectively. Therefore with the same assumption for the two tsunami sources, 'source 2' represents the first wave amplitude better than 'source 1'; but 'source 1' appears to perform slightly better in predicting the arriving time, 94% against 91%. Wei *et al.* (2014) also used the 'source 2' (tsunameter measurements) to initialise the MOST tsunami model and compared the results with those based on a GPS-tsunameter-derived source model and a USGS finite-fault

source model. Regardless of the advantages and limitations of these various real-time measurements with potential use in early tsunami warning, the tsunameter-derived source ('source 2') performed better in their research.

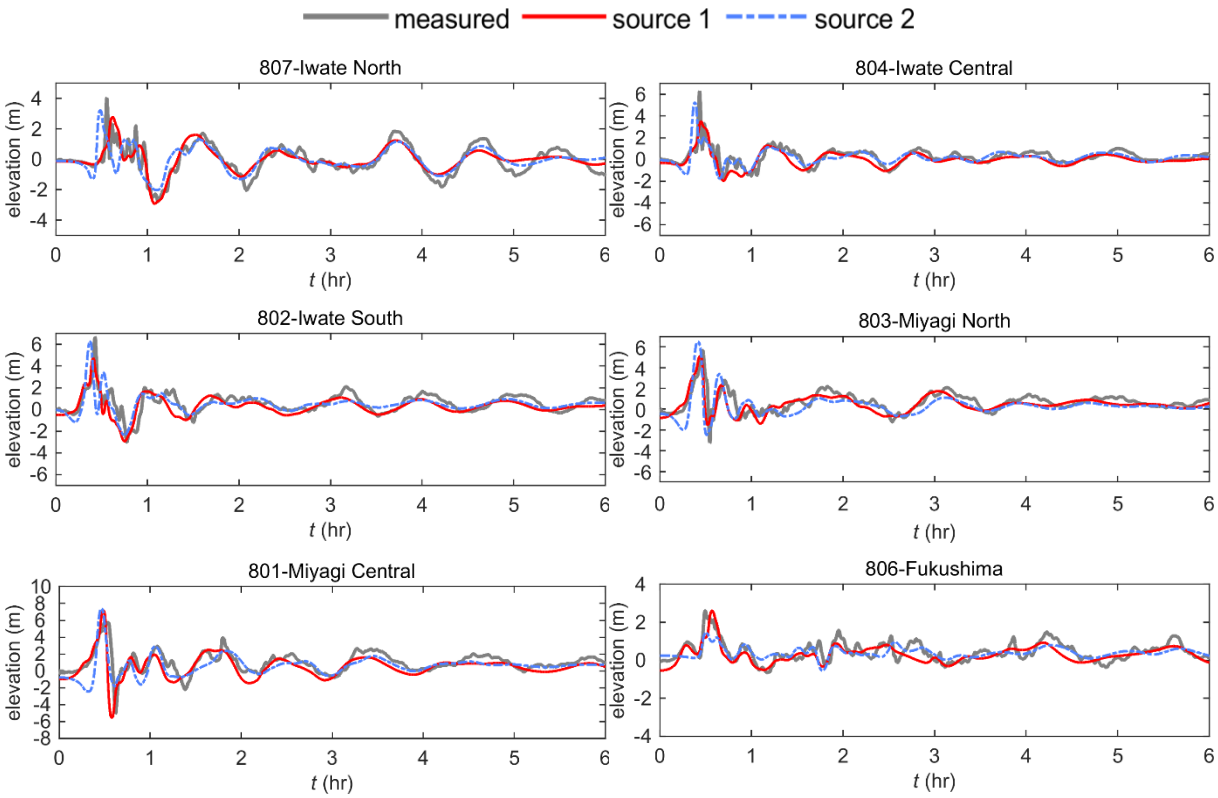


Figure 5.10: Model-observation comparison at 6 *GPS buoys* with two different source models.

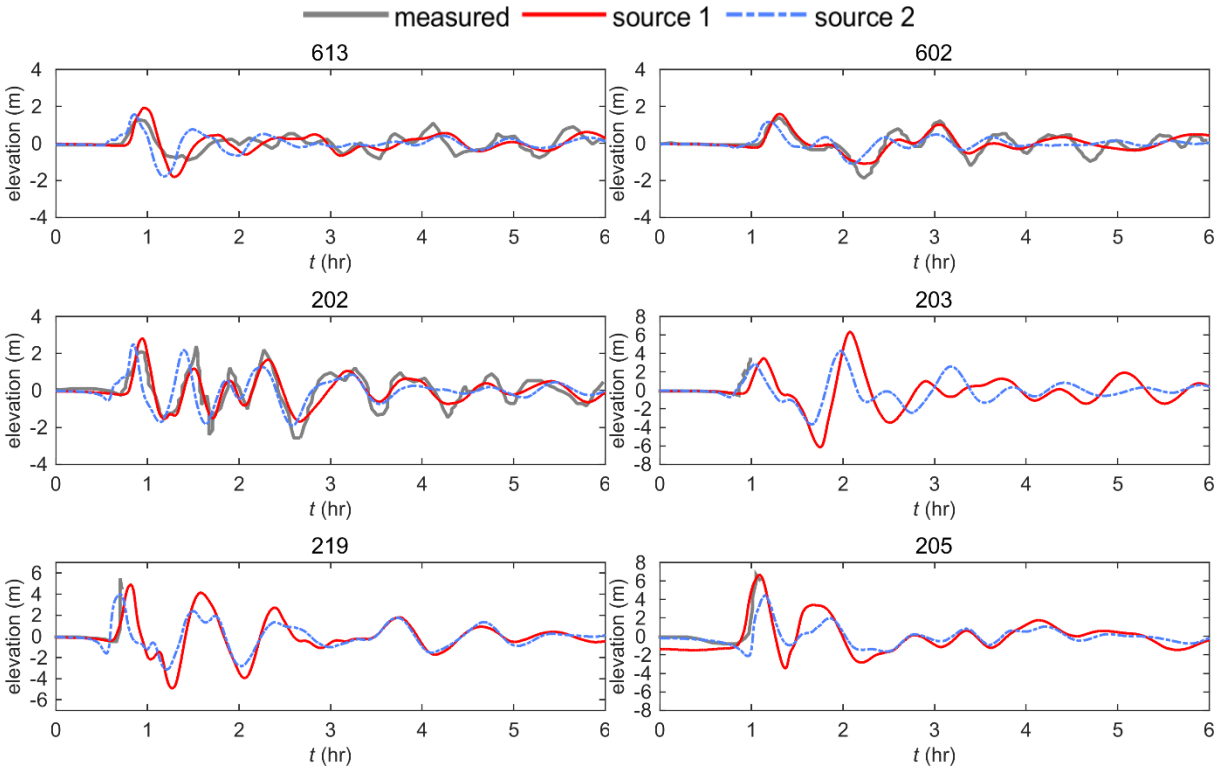


Figure 5.11: Model-observation comparison at 6 *coast gauges* with different two source models.

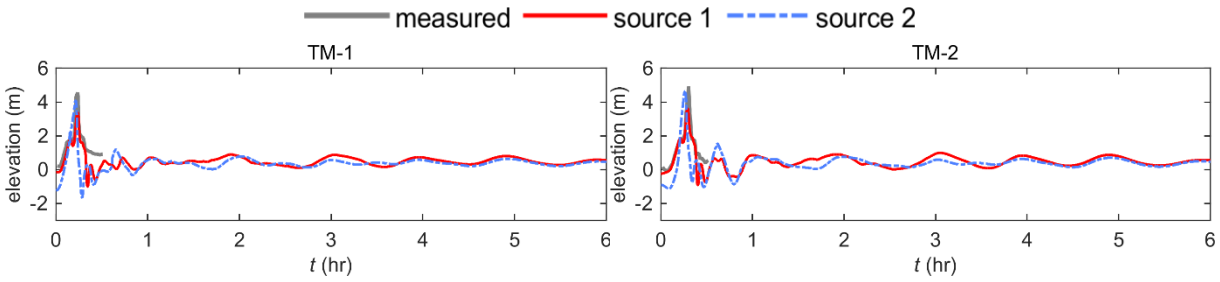


Figure 5.12: Model-observation comparison at two *bottom pressure gauges* with two different source models.

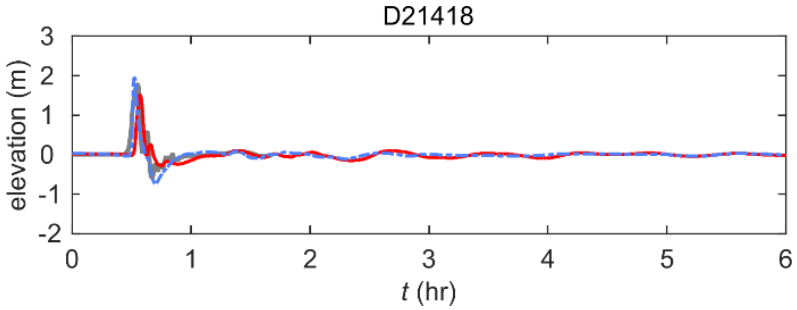


Figure 5.13: Model-observation comparison at *DART buoy 21418* with two different source models.

Station	Observed arrival time	Predicted arrival time and (E %)	Observed max. amp.	Predicted max. amp. and (E %)
807	33.3 min	37 min (+11.6 %)	4 m	2.77 m (-31 %)
804	26 min	27 min (+3.8 %)	6.3 m	3.51 m (-44.3 %)
802	25.6 min	24.6 min (-4 %)	6.7 m	4.7 m (-29.6 %)
803	28 min	26.5 min (-5.2 %)	5.7 m	4.93 m (-13.2 %)
801	30 min	29.2 min (-2.6 %)	5.8 m	6.65 m (+15.1 %)
806	29.6 min	34.1 min (+15.2 %)	2.6 m	2.6 m (0 %)
D21418	32 min	34 min (+6.3 %)	1.8 m	1.51 m (-16.1 %)
TM-1	14.3 min	14.2 min (-0.9 %)	5 m	3.36 m (-32.8 %)
TM-2	18.2 min	17.5 min (-4 %)	5 m	3.6 m (-28 %)
Model accuracy:		phase~94 %		amplitude ~76.6 %

Table 5.4: Comparison of the maximum tsunami amplitude and the arrival time of the leading wave between the model outputs based on 'source 1' and observations at the GPS buoys, deep ocean tsunameter and pressure gauges. The numbers in the brackets represent the relative error between the observations and model outputs.

Station	Observed arrival time	Predicted arrival time and (E %)	Observed max. amp. and (E %)	Predicted max. amp. and (E %)
807	33.3 min	29 min (-13 %)	4 m	3.22 m (-19.9 %)
804	26 min	22.9 min (-12 %)	6.3 m	5.25 m (-16.7 %)
802	25.6 min	22.3 min (-12.9 %)	6.7 m	6.27 m (-6 %)
803	28 min	26.1 min (-6.8 %)	5.7 m	6.19 m (+9 %)
801	30 min	28 min (-6.6 %)	5.8 m	6.88 m (+19 %)
806	29.6 min	30.2 min (+2.2 %)	2.6 m	1.46 m (-44.4 %)
D2148	32 min	31.4 min (-1.9%)	1.8 m	2 m (+11.1 %)
TM-1	14.3 min	12.9 min (-10.1 %)	5 m	4.08 m (-18.3 %)
TM-2	18.2 min	15.8 min (-13 %)	5 m	4.65 m (-7 %)
Model accuracy:		phase~91.3 %		amplitude ~ 83.14 %

Table 5.5: Comparison of the maximum tsunami amplitude and the arrival time of the leading wave between the model outputs based on 'source 2' and observations at the GPS buoys, deep ocean tsunameter and pressure gauges. The numbers in the brackets represent the relative error between the observations and model outputs.

5.4.4 *Effect of spatial resolution and the choice of slope limiter*

In this section the sensitivity of the hydrodynamic tsunami model to different spatial resolutions and the choice of different slope limiters is studied. For the 2011 Japan tsunami induced by the Tohoku earthquake, the focus of sensitivity analysis in the literature is more on the choice of source models. For example, Goda *et al.* (2014) conducted a rigorous sensitivity analysis of tsunami wave profiles concerning the uncertainty of earthquake slip and fault geometry. The results presented here show that the choice of slope limiters in the finite volume scheme and the spatial resolution used in the simulations can also have significant impact on the simulation results when the same source model is used. Simulations are carried out with different spatial resolutions, including 450 m, 900 m, 1350 m, 1800 m, and 3600 m. The slope limiters used include the minmod (Roe, 1986), Sweby (Sweby, 1984), and Superbee (Roe, 1986), which have been introduced in (Chapter 3-Subsection 3.1.3). The first-order (FO) accurate scheme, as the most dissipative model, is also included for a comparison. A total of 20 simulations have been carried out and the results for 'source 1' and 'source 2' are shown in (Figure 5.14-5.24, Pages 113-123) and (Figure 5.25-5.35, Pages 124-134), respectively, in comparison with observations at 11 selected gauges. Firstly the effect of space resolution is discussed under the same limiter. It is clear from the results that simulations based on very coarse resolutions lead to significant numerical damping of the tsunami wave both on the lead wave and trailing waves for most gauges; but the spatial resolutions seem to only have a minor effect on the phase.

This phenomenon is evident at the simulation with a resolution of 3600 m. The damping effect caused by the coarse resolution is more due to numerical dissipation as the wave travels a long distance in the deep ocean. Wang and Liu (2006) and Popinet (2011) also showed a significant underestimation of wave elevation for the 2004 Indian ocean tsunami when using coarse spatial resolutions in their simulations.

Another source of numerical damping which seems to be less discussed or studied in the context of tsunami modelling is the use of slope limiters in the second-order Godunov scheme. These limiters are used in high-order accurate numerical schemes to avoid spurious oscillations near the numerical solutions containing shocks and flow discontinuities, and to provide the so-called total variation diminishing (TVD) solutions. As illustrated in Figure 5.14-5.35 (Pages 113-134), by using the same spatial resolution across the simulations the results show that the Sweby limiter can better capture higher wave peaks compared to the minmod limiter and this is more evident especially for the leading waves with higher amplitudes. This is explained by the fact that the mimod limiter is the most dissipative of the TVD limiters. On the other hand the Superbee limiter is the least dissipative limiter which may cause numerical instability. In this regard, the use of Superbee limiter at the simulation with 450 m resolution leads to spurious oscillations, confirming that this limiter may cause numerical instability. The results are not included in the comparison for this simulation (i.e. Superbee limiter + 450 m resolution). The simulation with the first-order scheme produces the most dissipative waveform at a given spatial resolution and therefore the use of the first-order scheme leads to the most dissipative results in the coarse-resolution simulations.

To quantitatively evaluate the simulation sensitivity associated with different slope limiters and spatial resolutions, the relative errors between predicted and measured maximum wave amplitude are calculated. Table 5.6 (Pages 135-136) presents the relative errors calculated between the numerical predictions based on 'source 2' and the observations at 6 GPS buoys 807, 804, 802, 803, 801 and 806. The mean absolute error of these GPS buoys obtained using different resolutions and slope limiters are presented in Table 5.7 (Page 137). At fine resolutions (less than 1 km), the minmod limiter can reproduce the wave peak close to observation. Simulations with coarser resolutions (up to 1800 m) with the Sweby limiter can also predict the wave amplitude properly. It is found that the simulation on a fine resolution with minmod limiter is roughly equivalent to the simulation by halving the spatial resolution but using the Sweby limiter. The results obtained from the Sweby limited simulations with coarse resolutions (e.g. between 1-2 km) are satisfactory and the combination is recommended for the practical applications as the computational time can be reduced significantly compared to the finer resolution simulations with a minmod limiter.

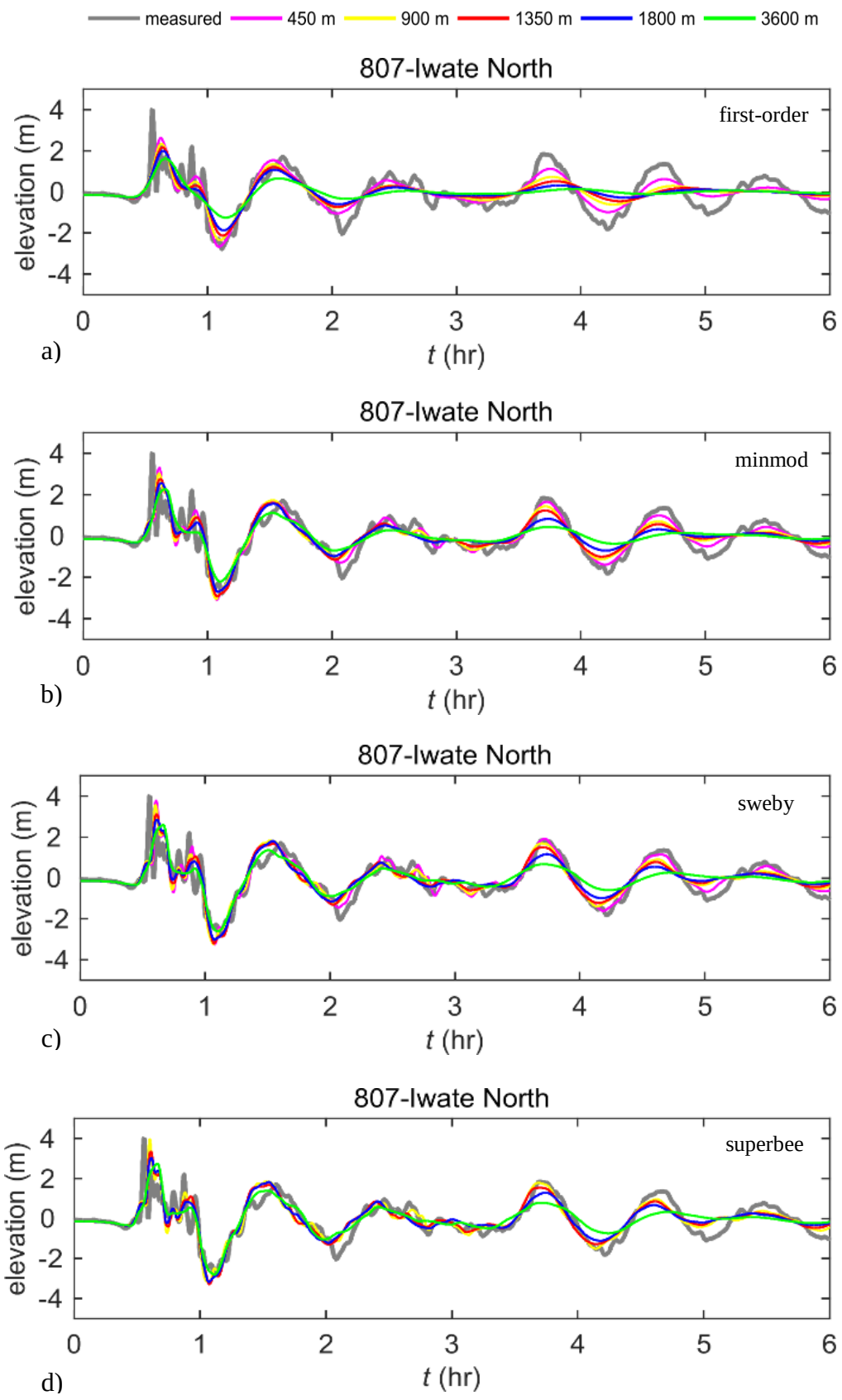


Figure 5.14: The sensitivity analysis results considering different resolutions and slope limiters for 'source 1' at gauge 807, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

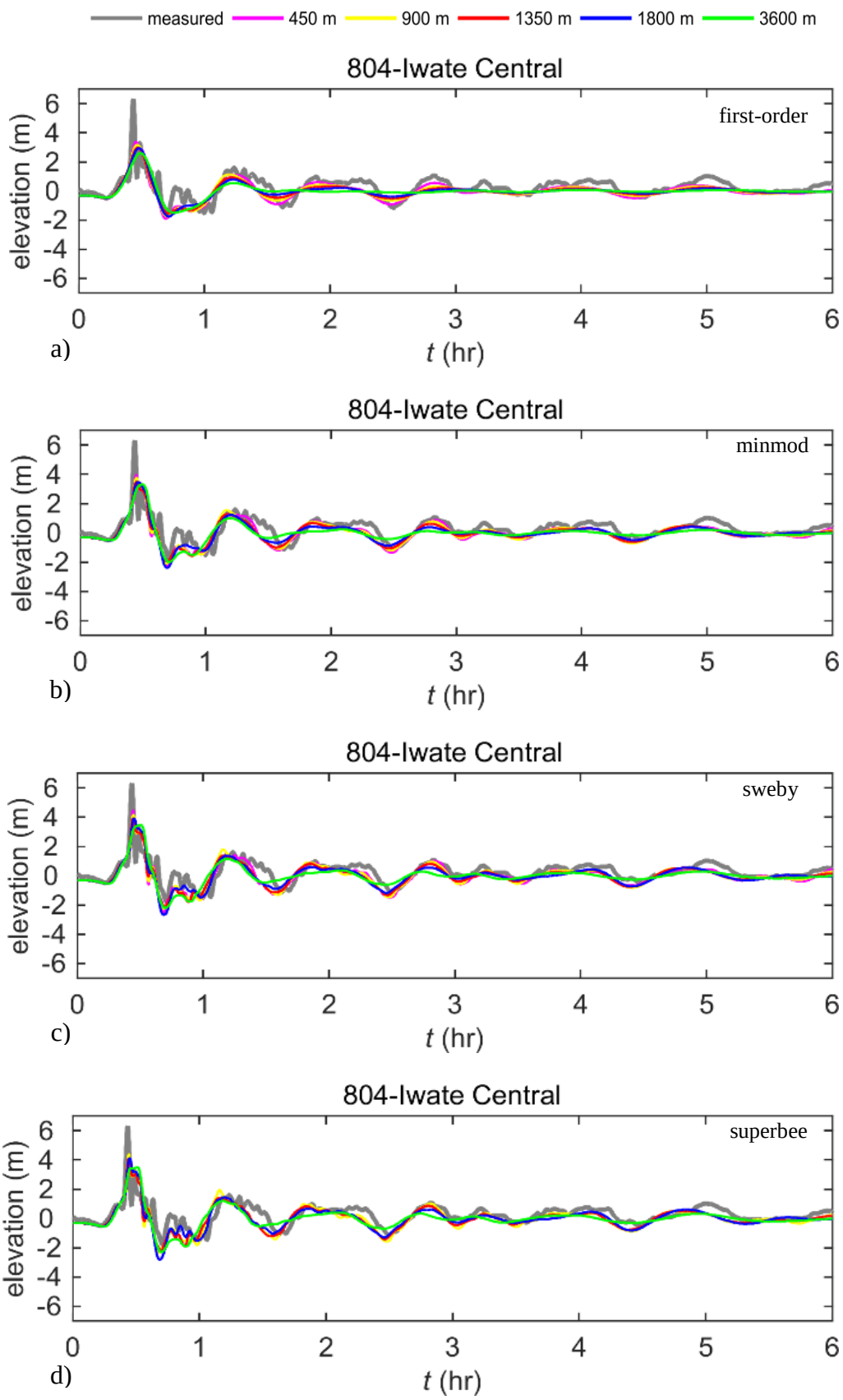


Figure 5.15: The sensitivity analysis results considering different resolutions and slope limiters for 'source 1' at gauge 804, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

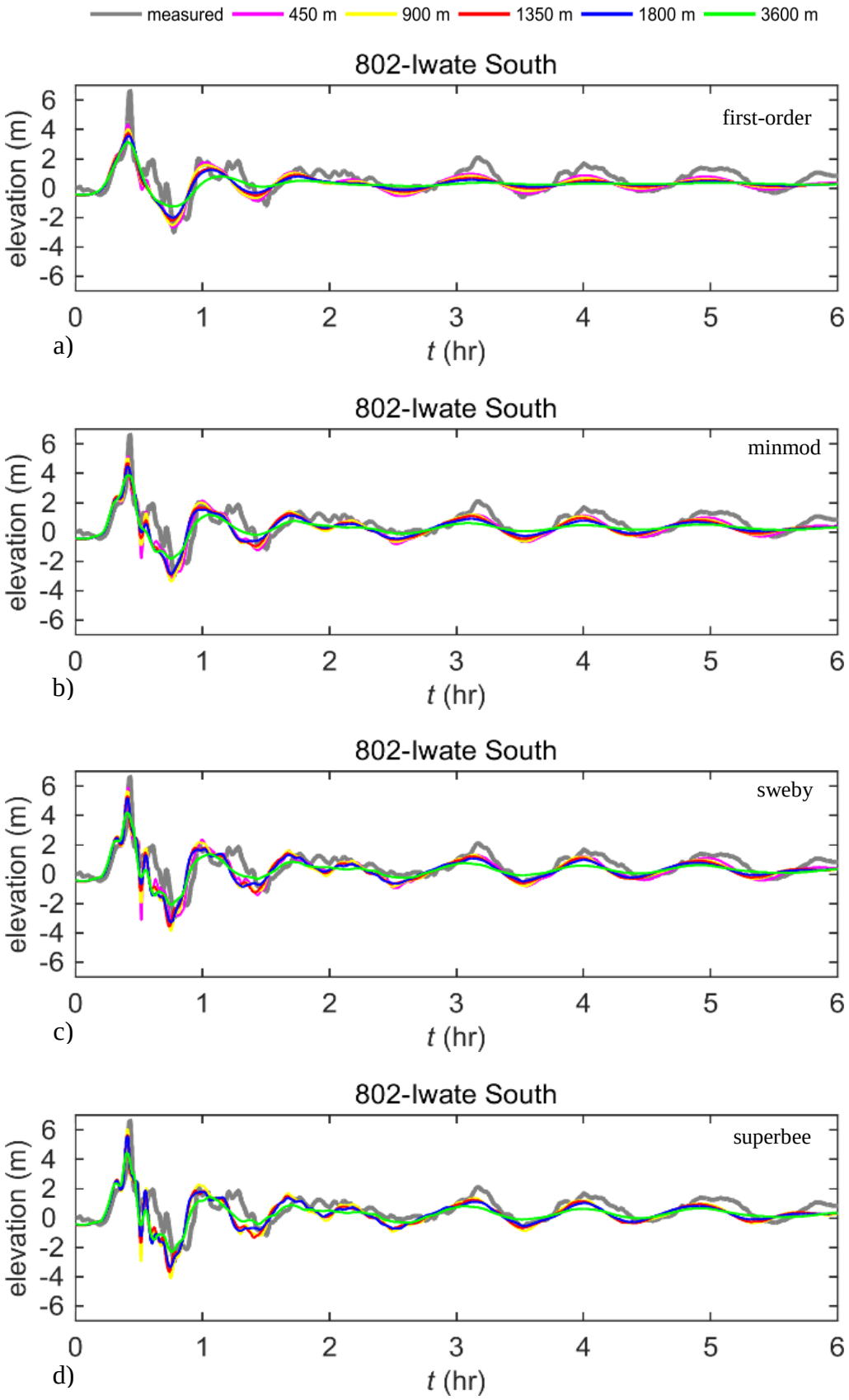


Figure 5.16: The sensitivity analysis results considering different resolutions and slope limiters for 'source 1' at gauge 802, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

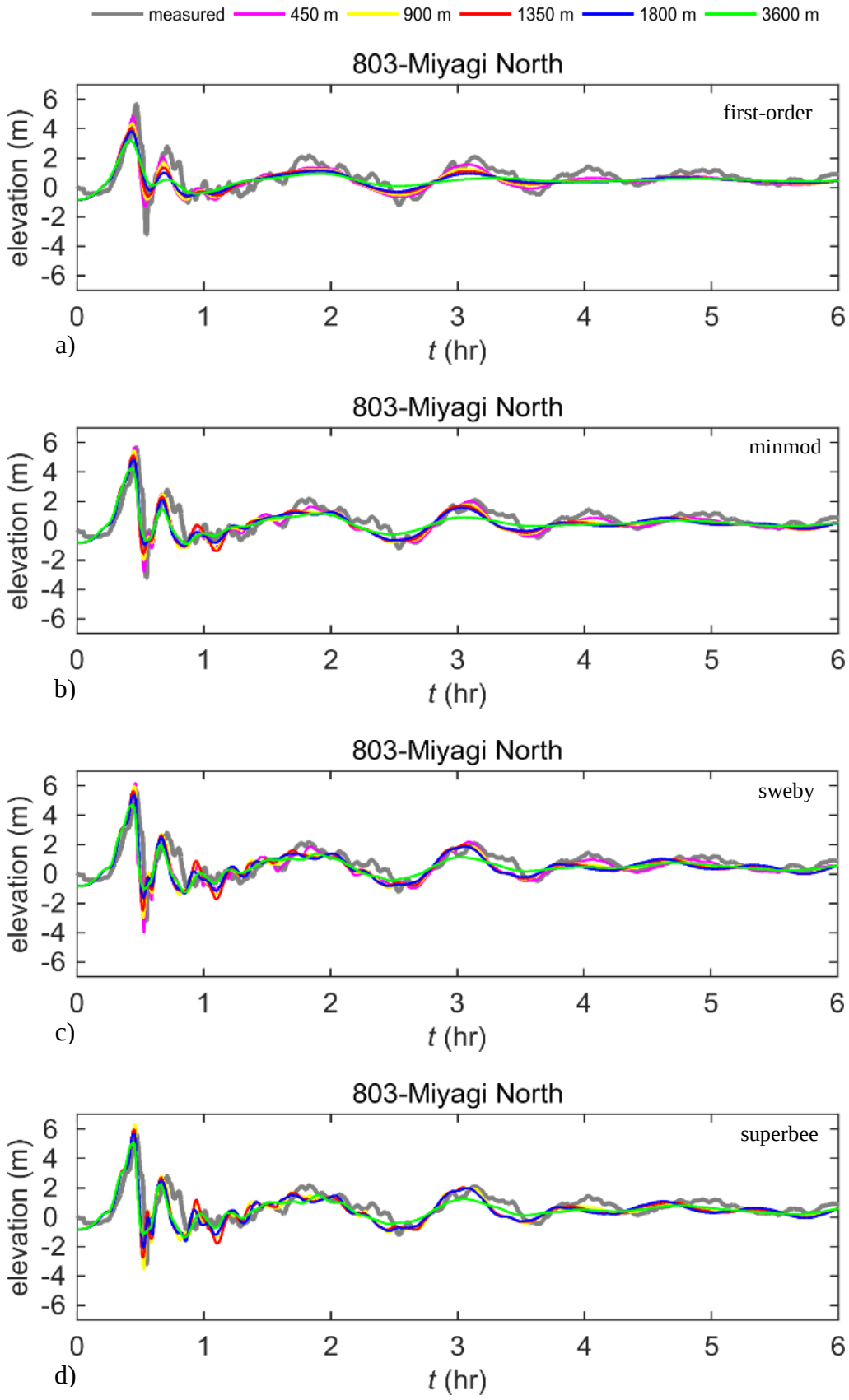


Figure 5.17: The sensitivity analysis results considering different resolutions and slope limiters for 'source 1' at gauge 803, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

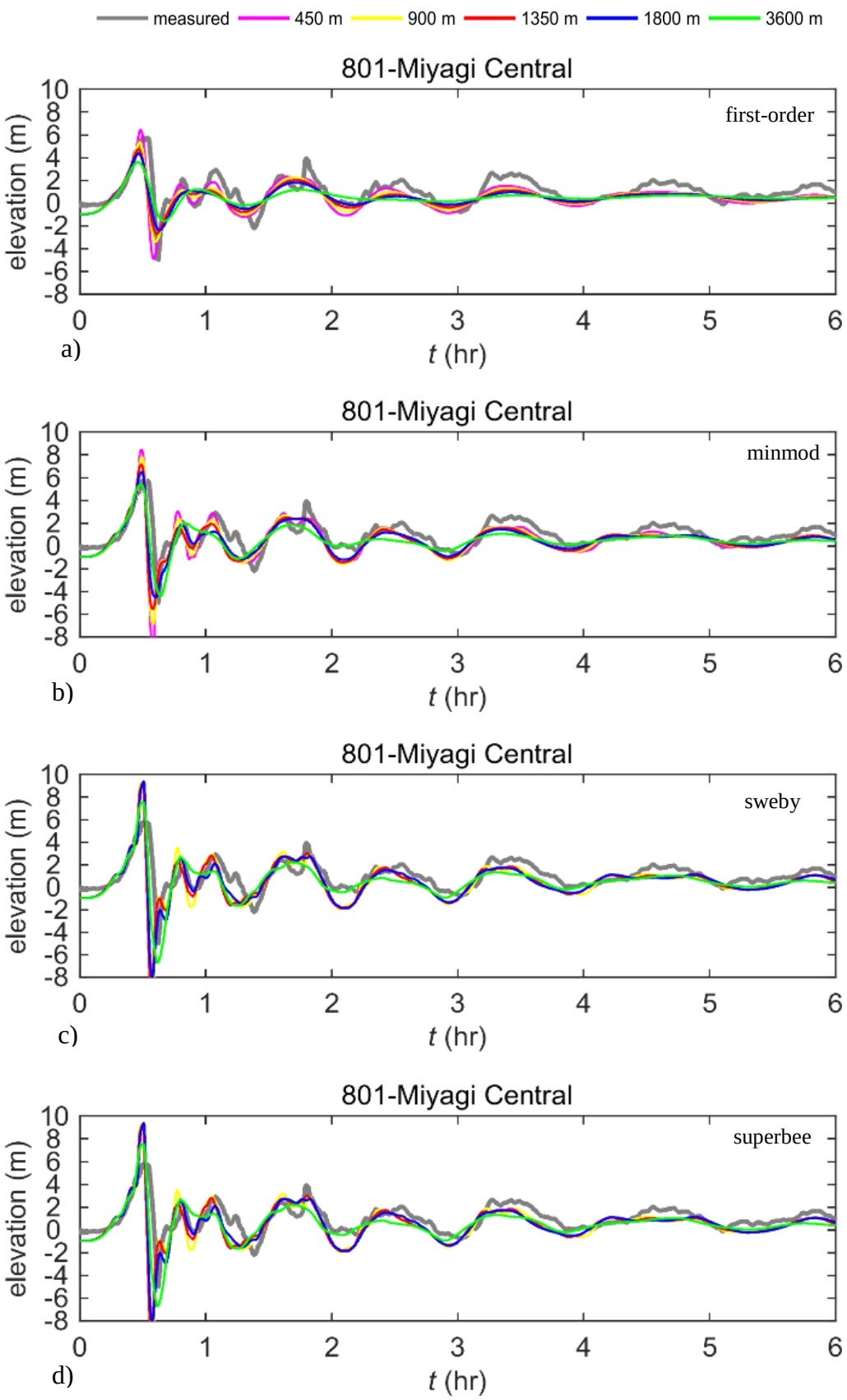


Figure 5.18: The sensitivity analysis results considering different resolutions and slope limiters for 'source 1' at gauge 801, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

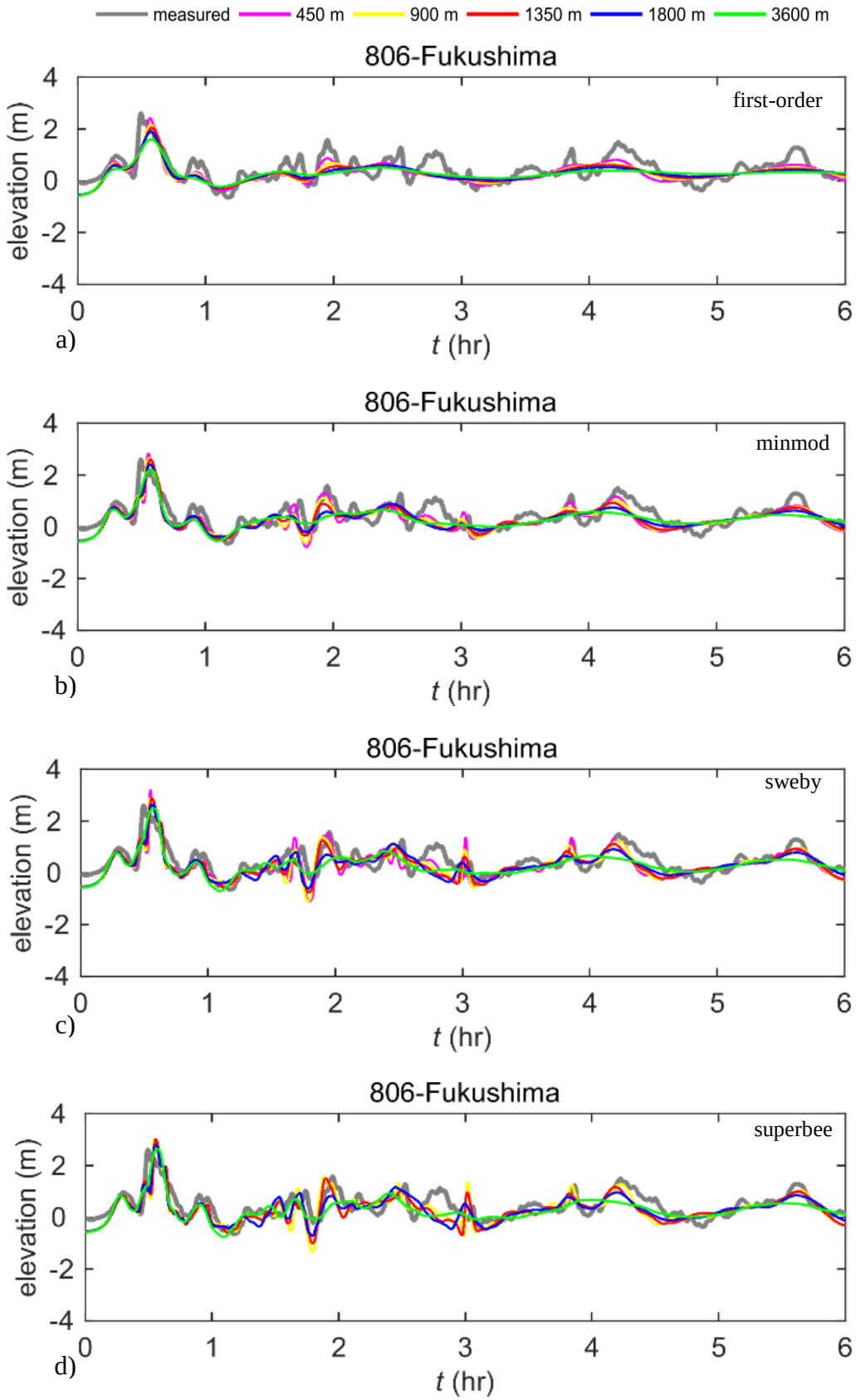


Figure 5.19: The sensitivity analysis results considering different resolutions and slope limiters for 'source 1' at gauge 806, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

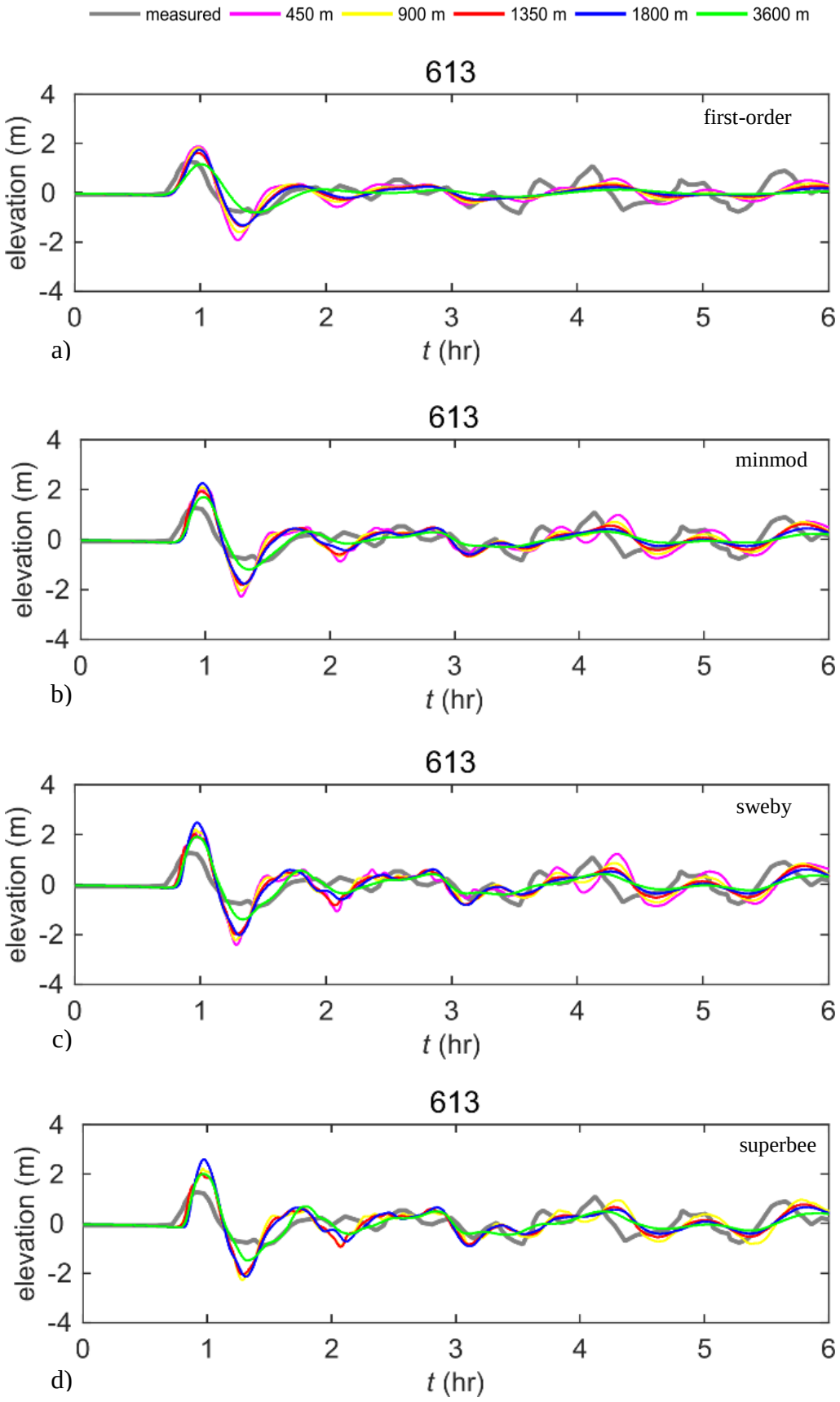


Figure 5.20: The sensitivity analysis results considering different resolutions and slope limiters for 'source 1' at gauge 613, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

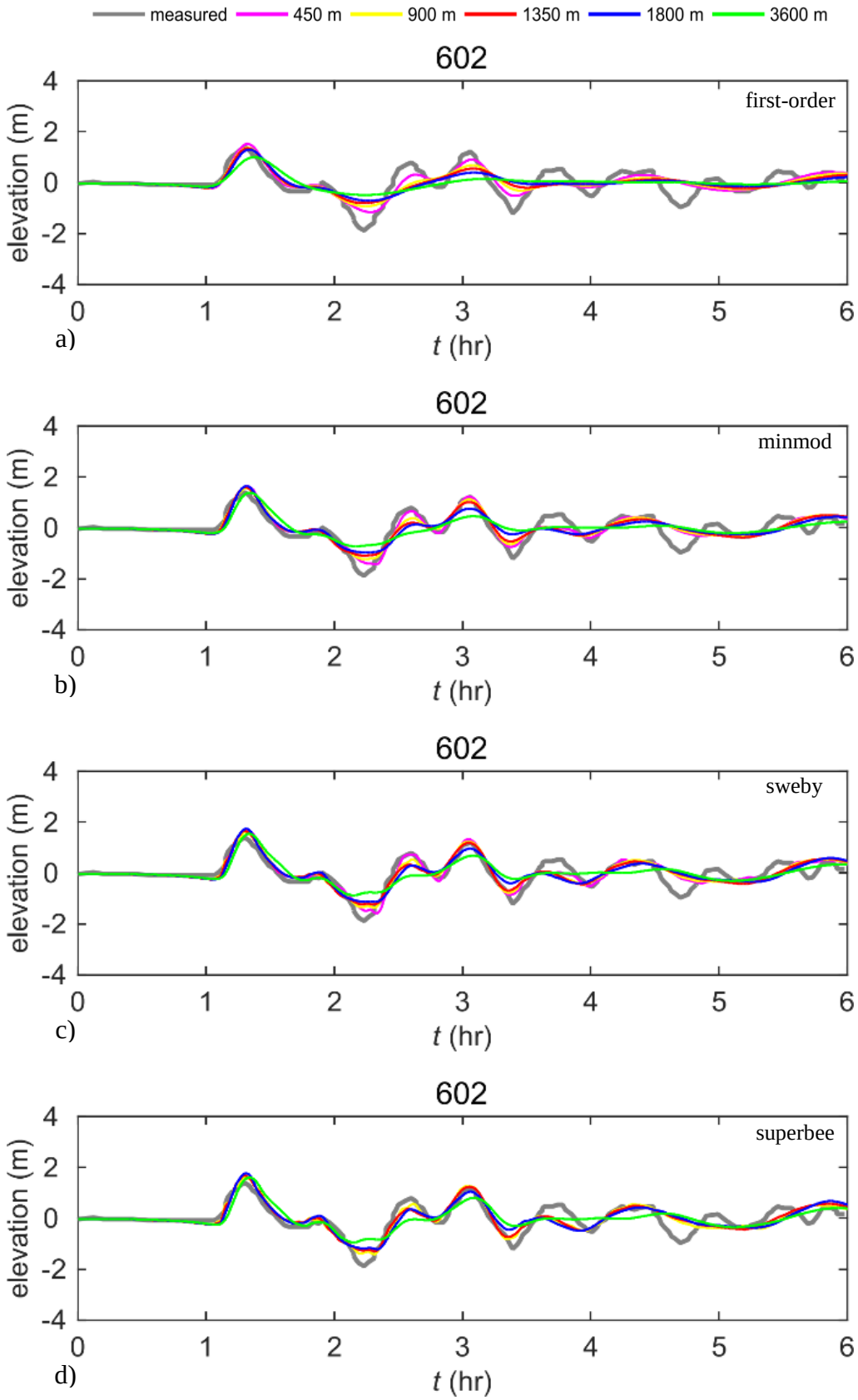


Figure 5.21: The sensitivity analysis results considering different resolutions and slope limiters for 'source 1' at gauge 602, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

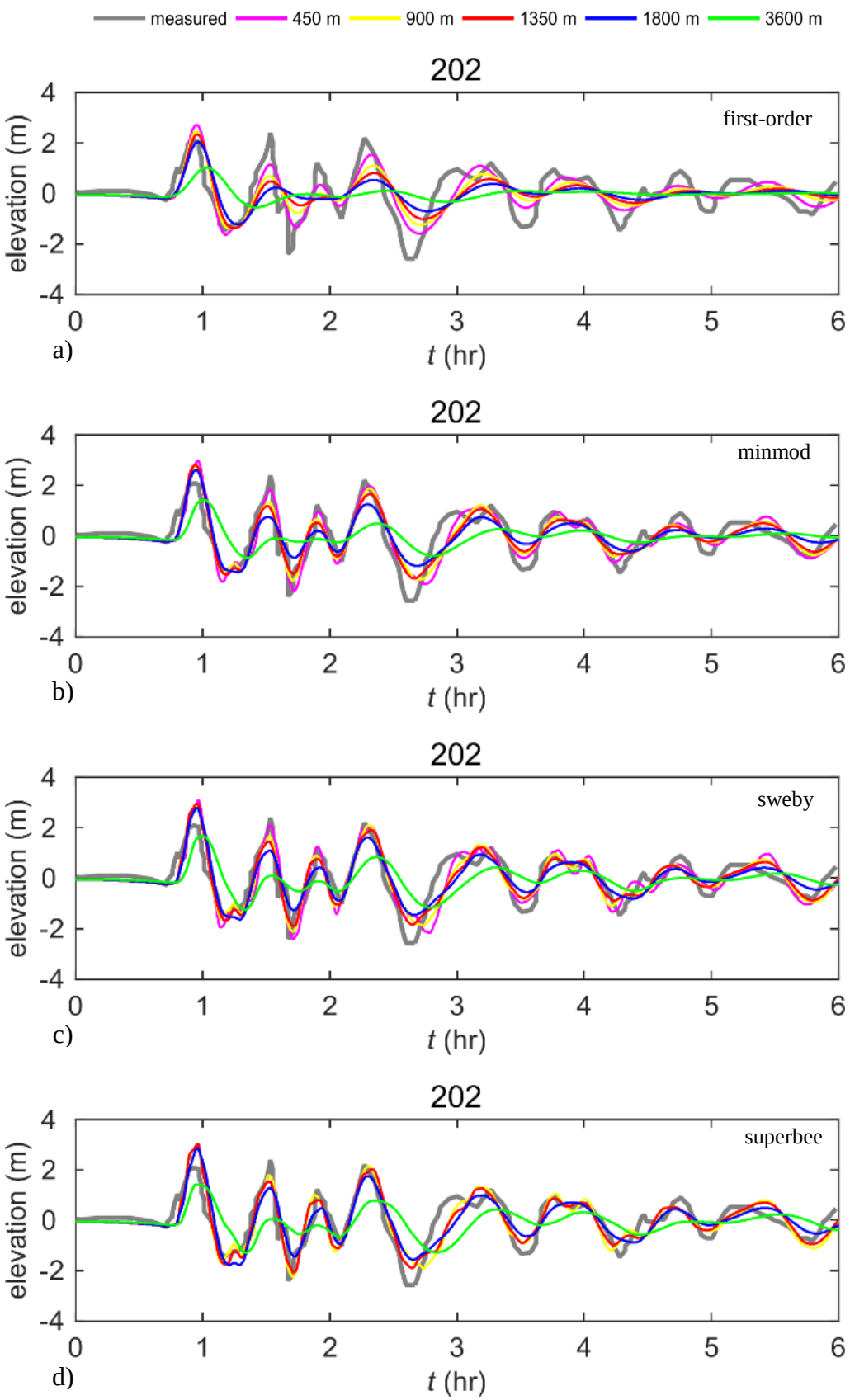


Figure 5.22: The sensitivity analysis results considering different resolutions and slope limiters for 'source 1' at gauge 202, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

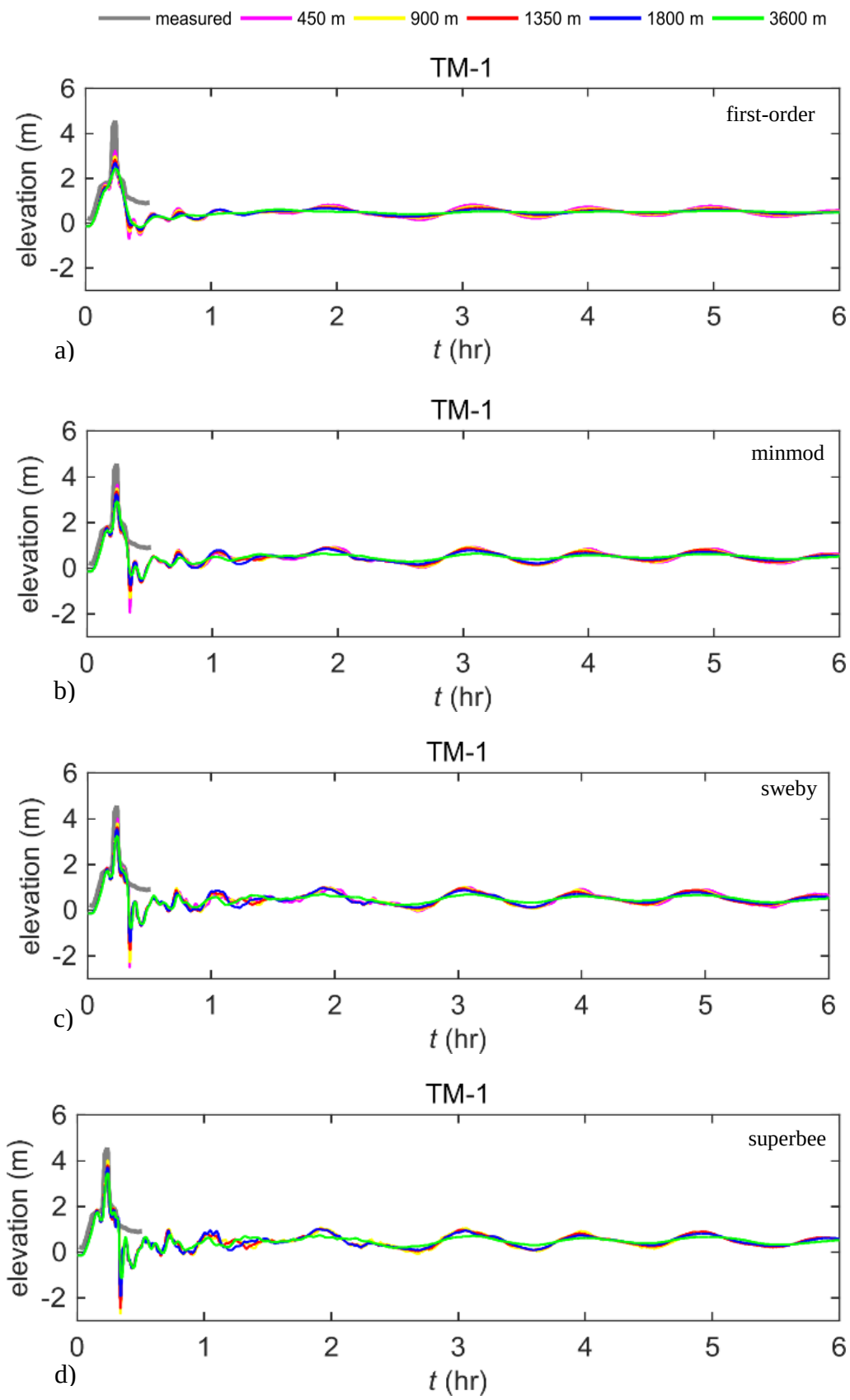


Figure 5.23: The sensitivity analysis results considering different resolutions and slope limiters for 'source 1' at gauge TM-1, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

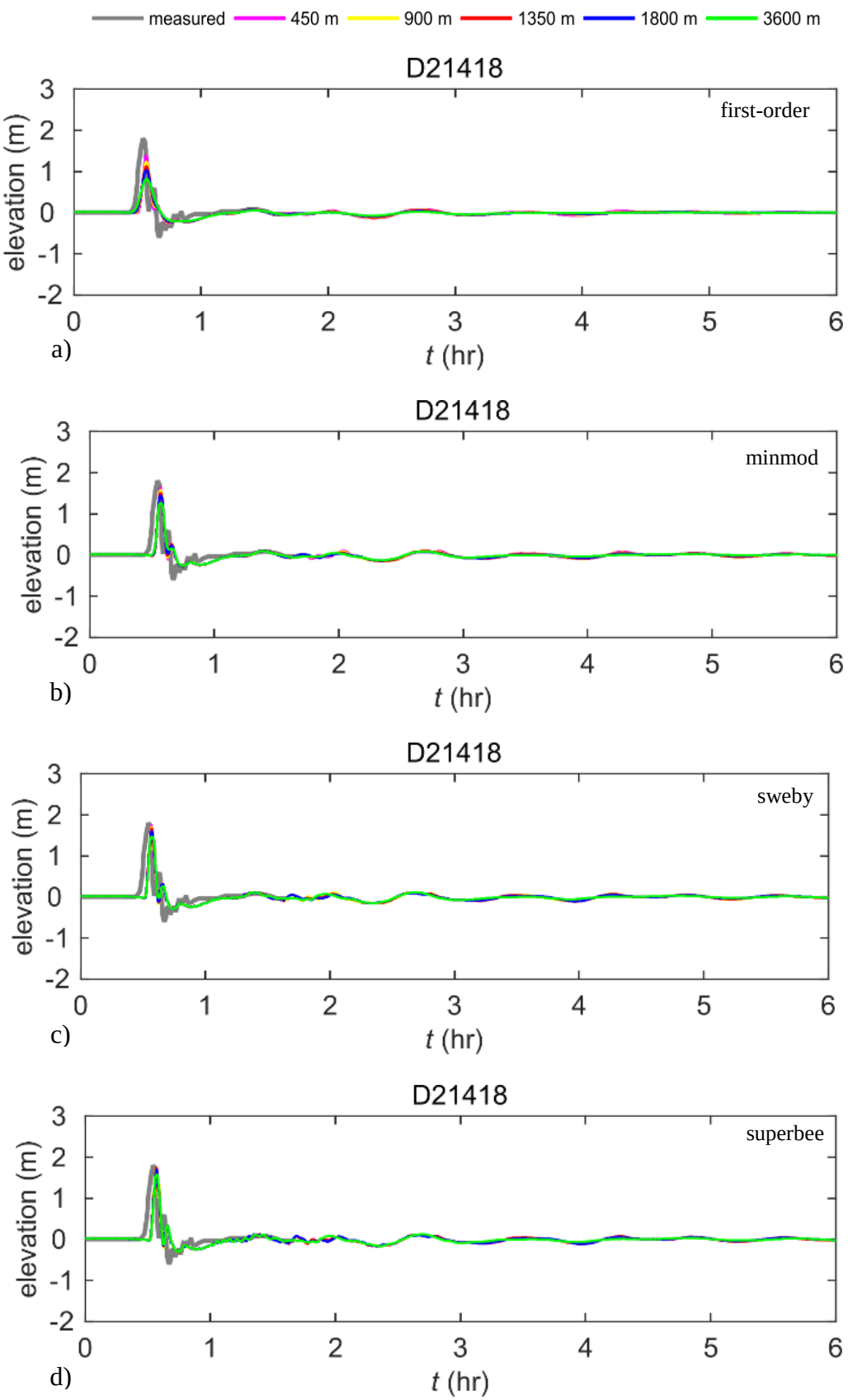


Figure 5.24: The sensitivity analysis results considering different resolutions and slope limiters for 'source 1' at gauge D21418, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

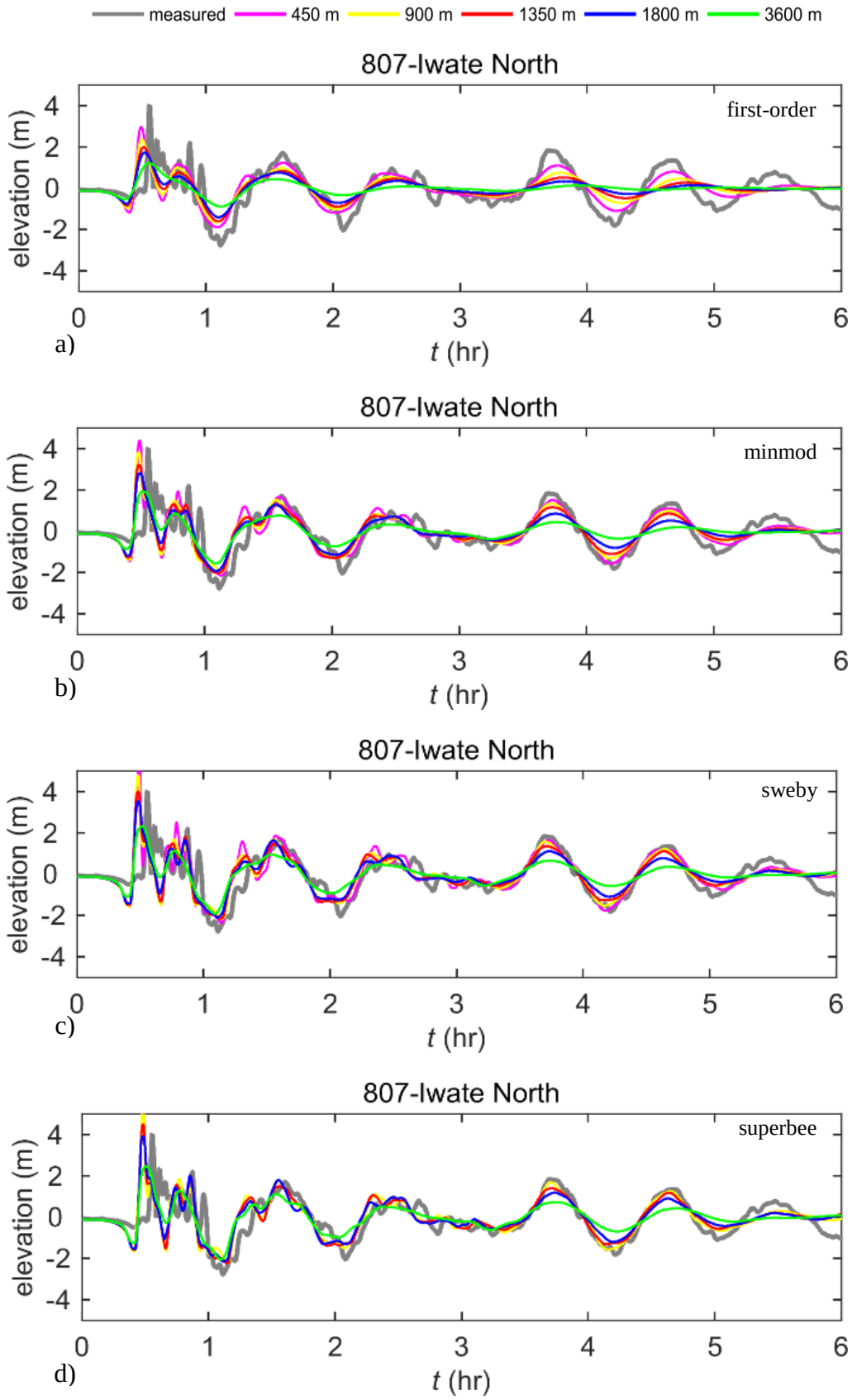


Figure 5.25: The sensitivity analysis results considering different resolutions and slope limiters for 'source 2' at gauge 807, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

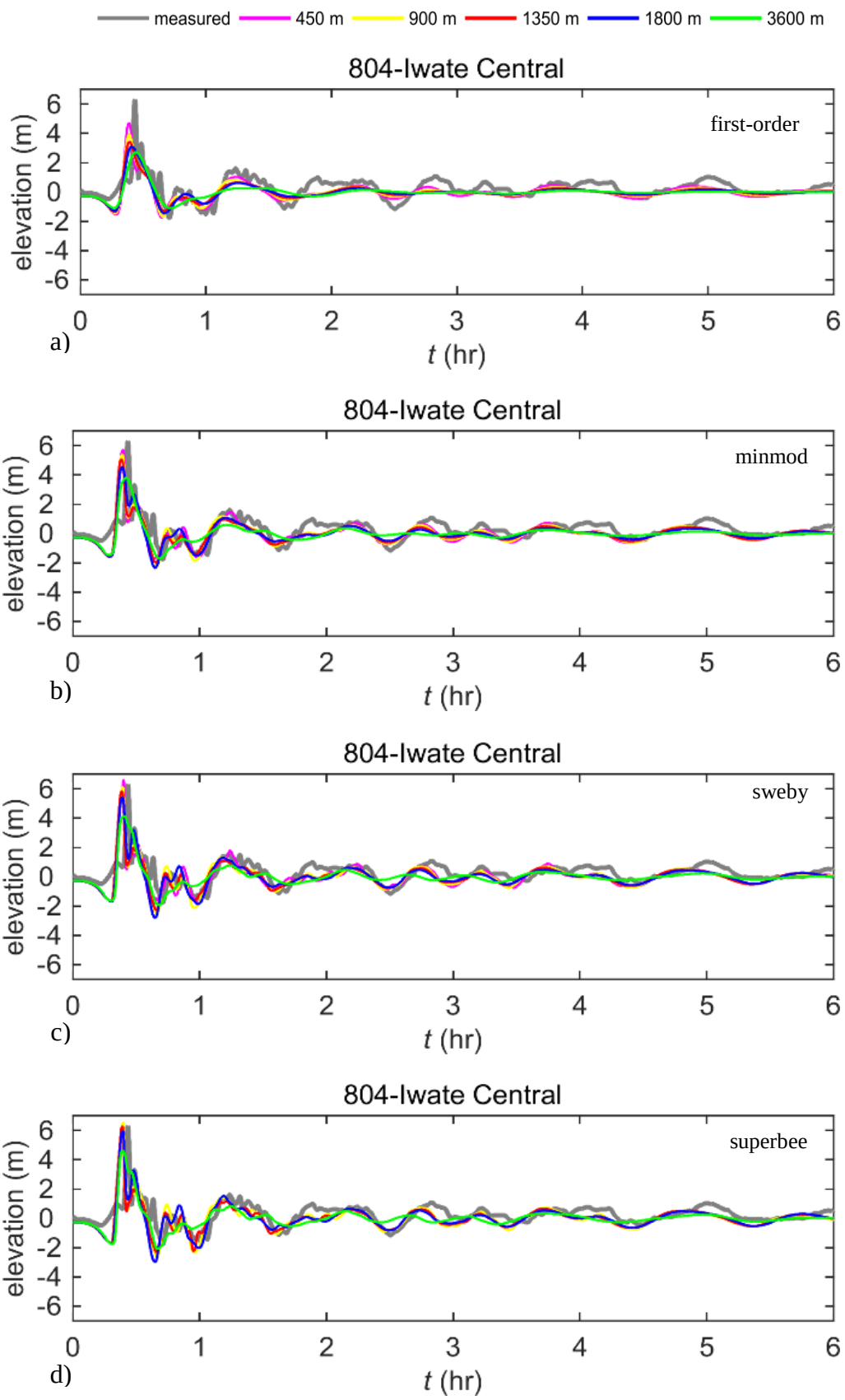


Figure 5.26: The sensitivity analysis results considering different resolutions and slope limiters for 'source 2' at gauge 804, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

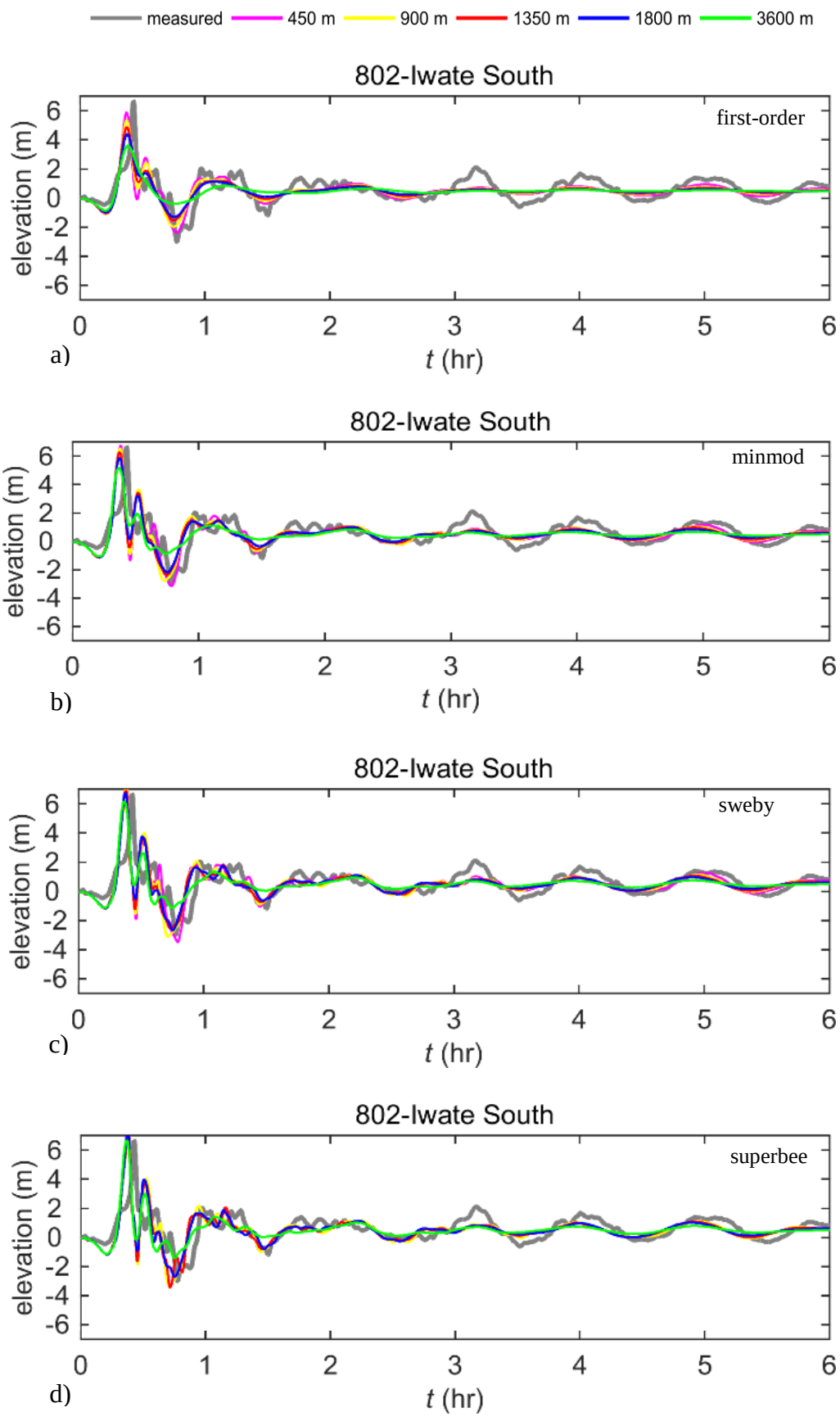


Figure 5.27: The sensitivity analysis results considering different resolutions and slope limiters for 'source 2' at gauge 802, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

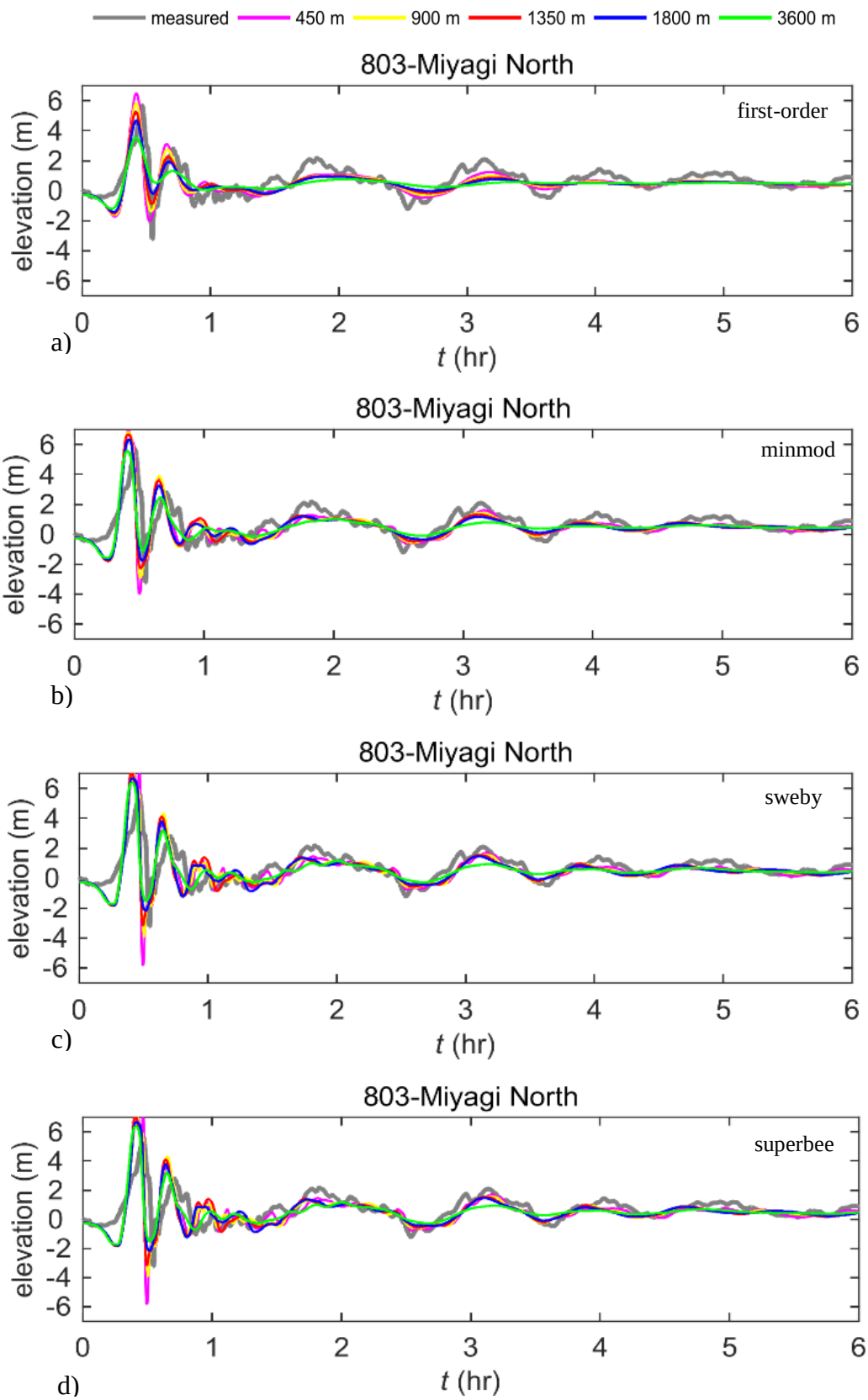


Figure 5.28: The sensitivity analysis results considering different resolutions and slope limiters for 'source 2' at gauge 803, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

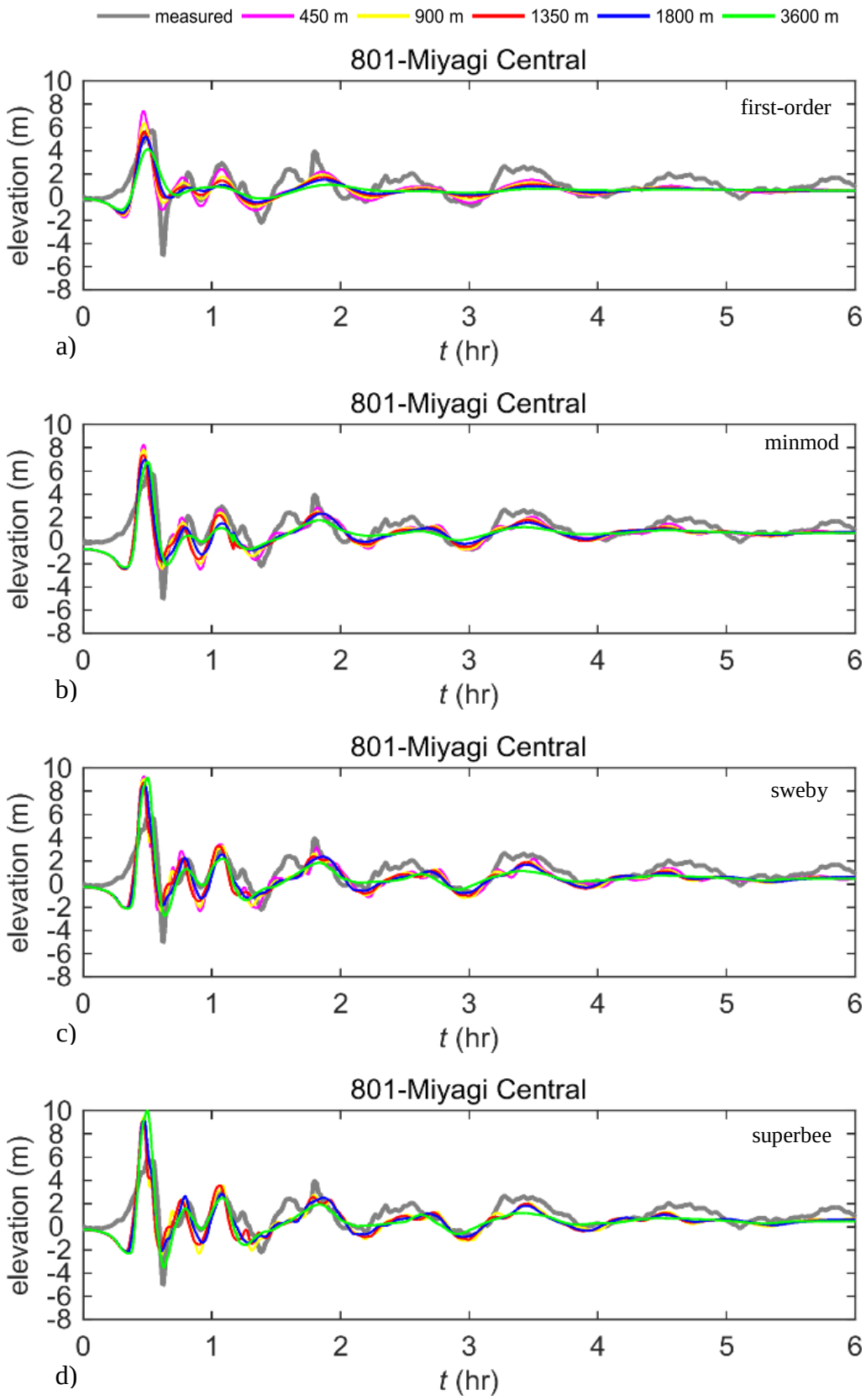


Figure 5.29: The sensitivity analysis results considering different resolutions and slope limiters for 'source 2' at gauge 801, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

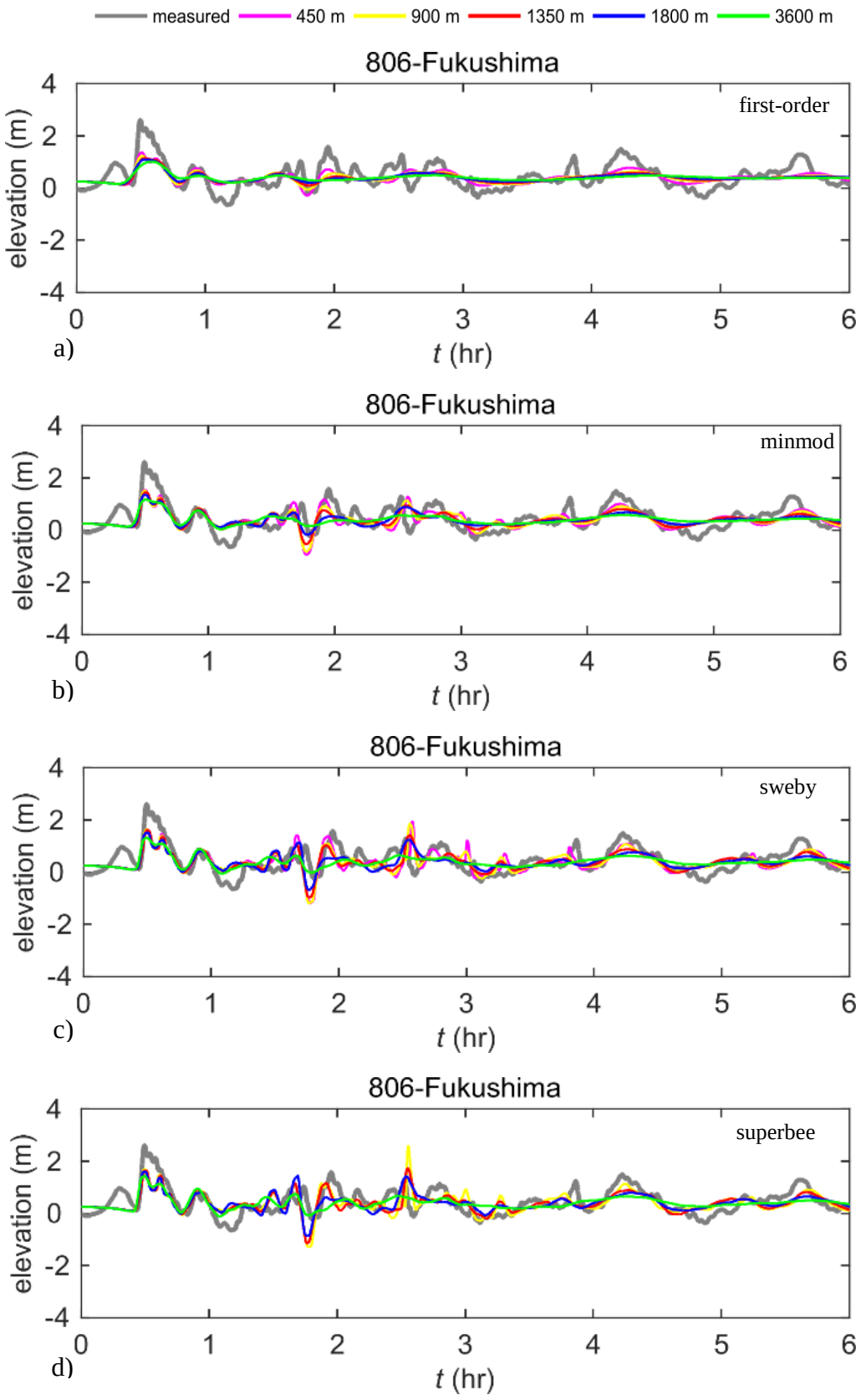


Figure 5.30: The sensitivity analysis results considering different resolutions and slope limiters for 'source 2' at gauge 806, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

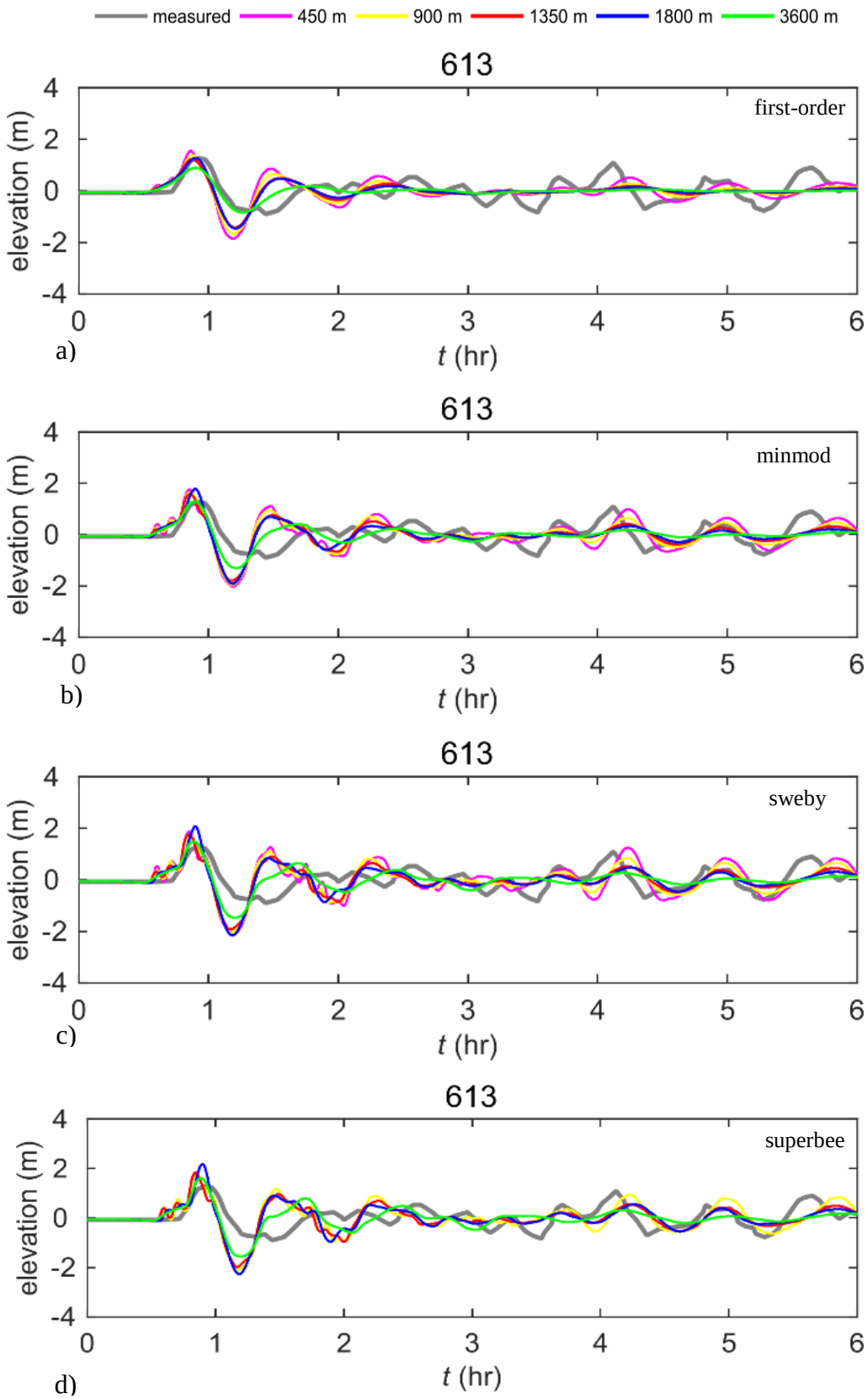


Figure 5.31: The sensitivity analysis results considering different resolutions and slope limiters for 'source 2' at gauge 613, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

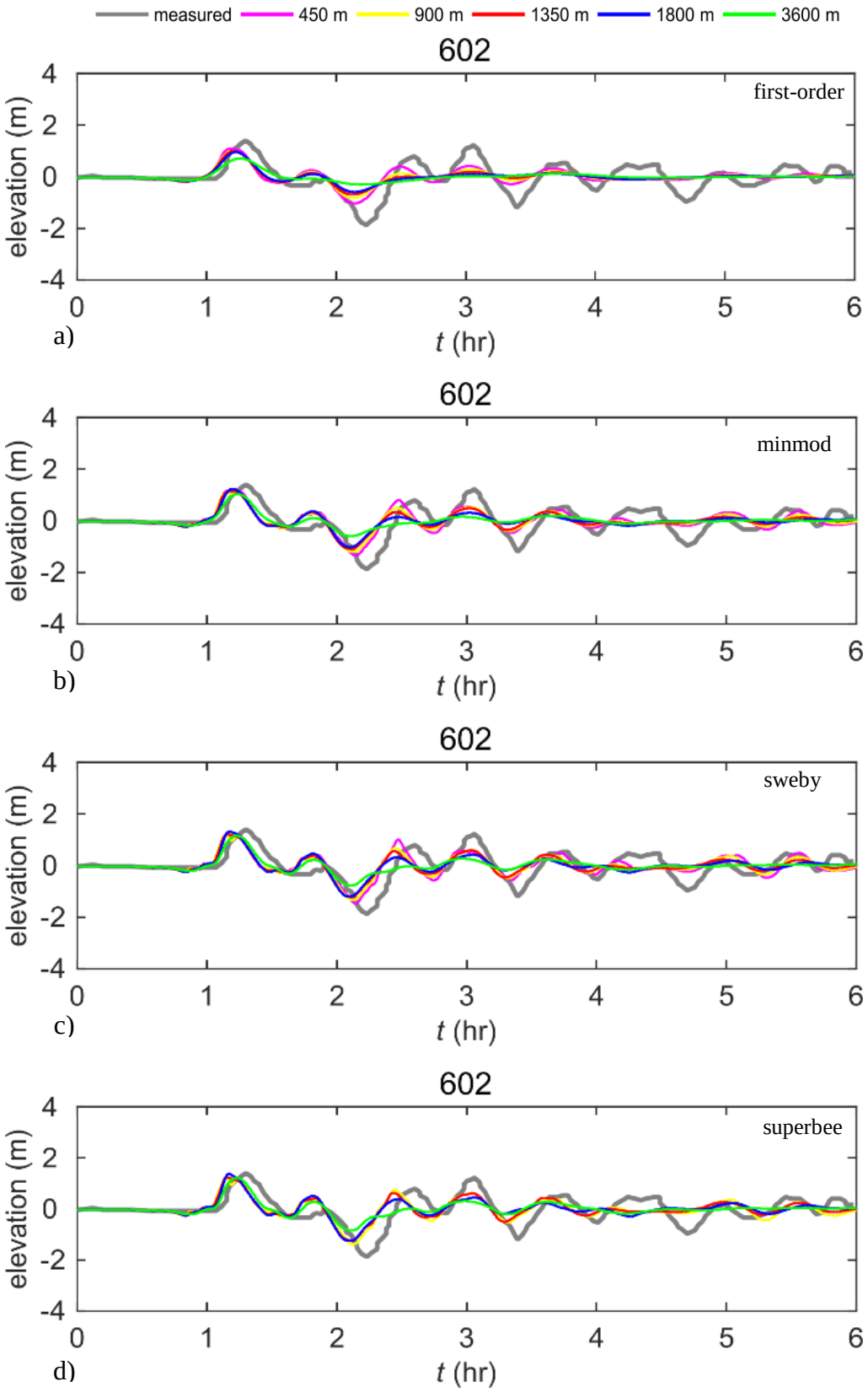


Figure 5.32: The sensitivity analysis results considering different resolutions and slope limiters for 'source 2' at gauge 602, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

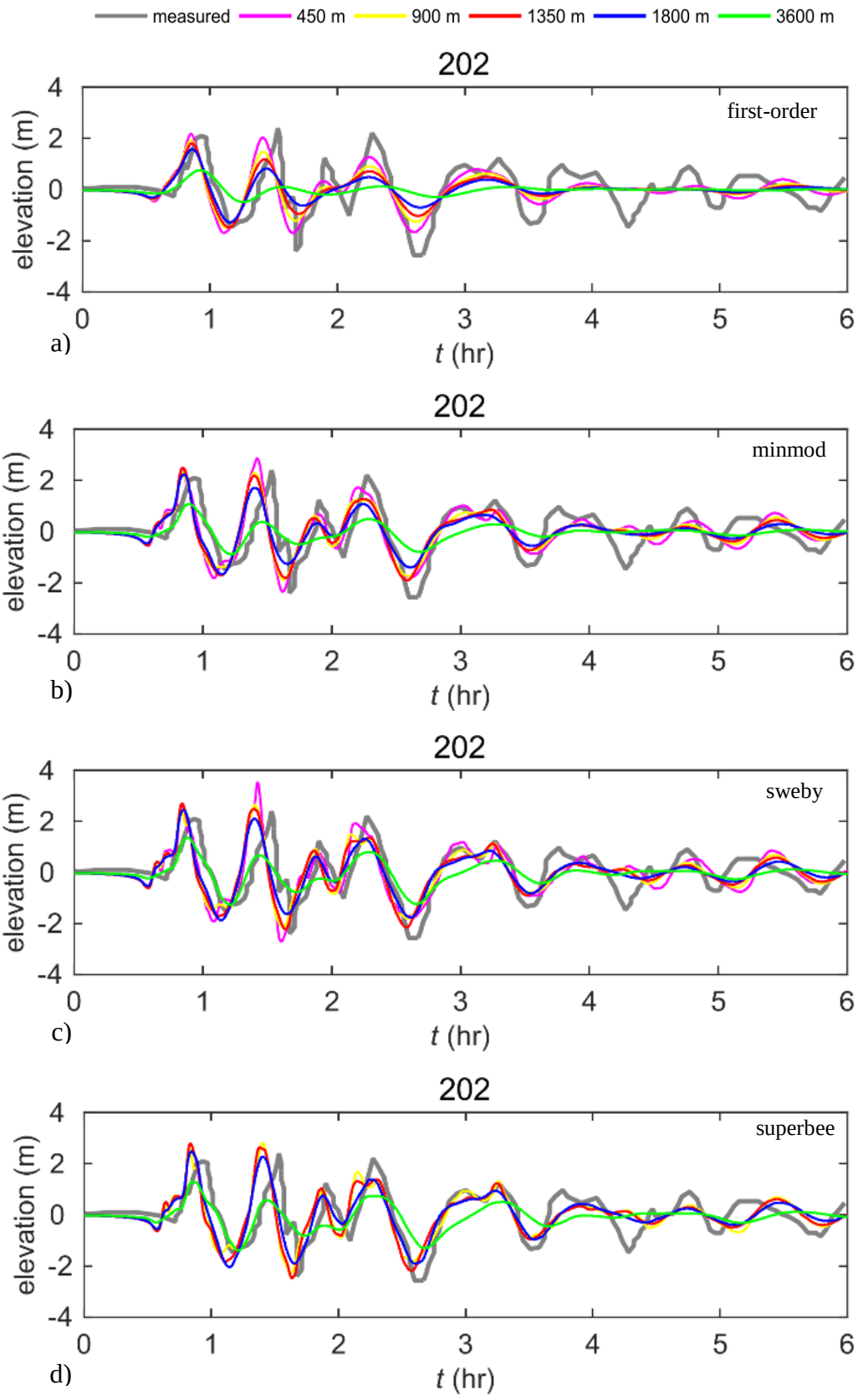


Figure 5.33: The sensitivity analysis results considering different resolutions and slope limiters for 'source 2' at gauge 202, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

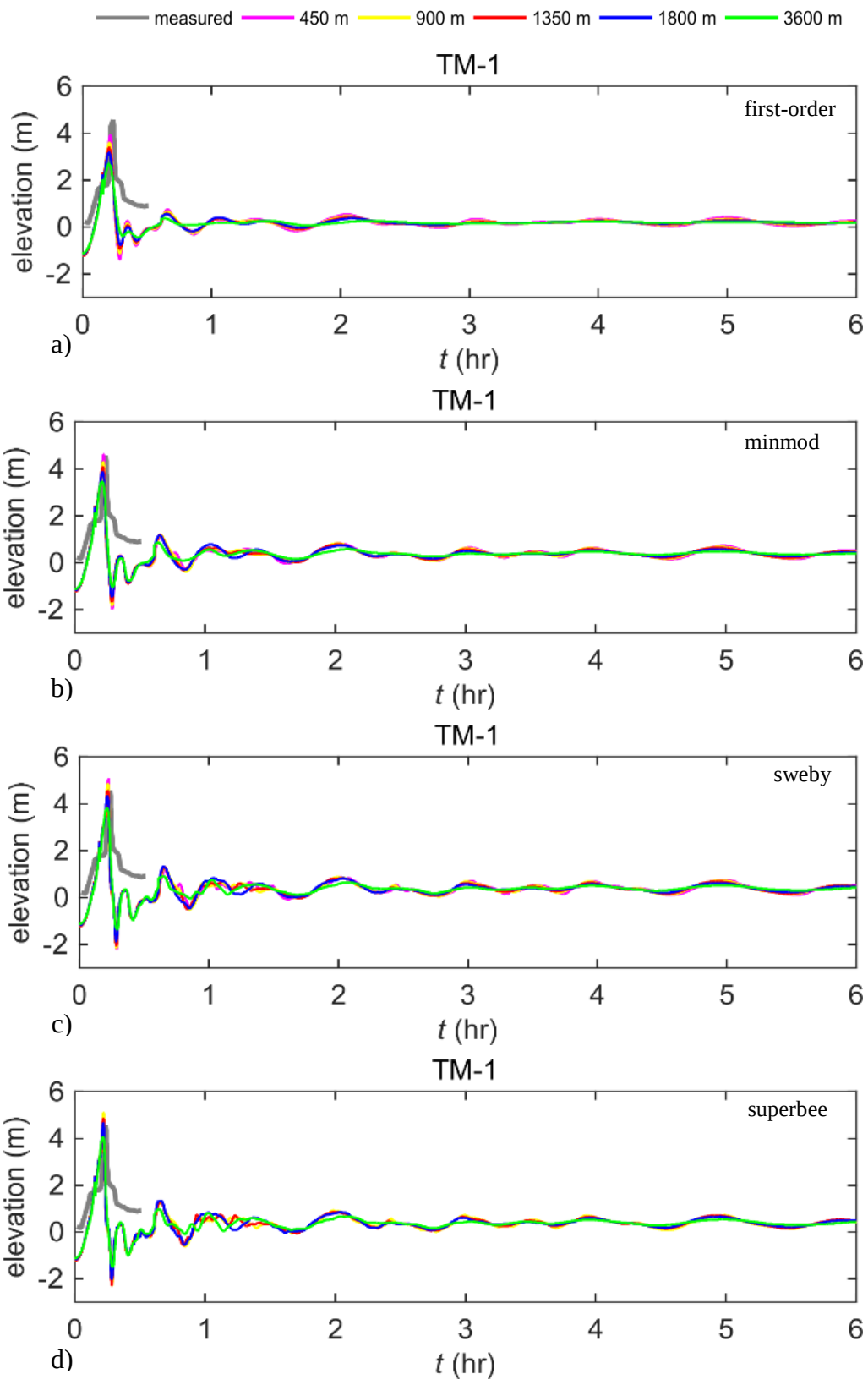


Figure 5.34: The sensitivity analysis results considering different resolutions and slope limiters for 'source 2' at gauge TM-1, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

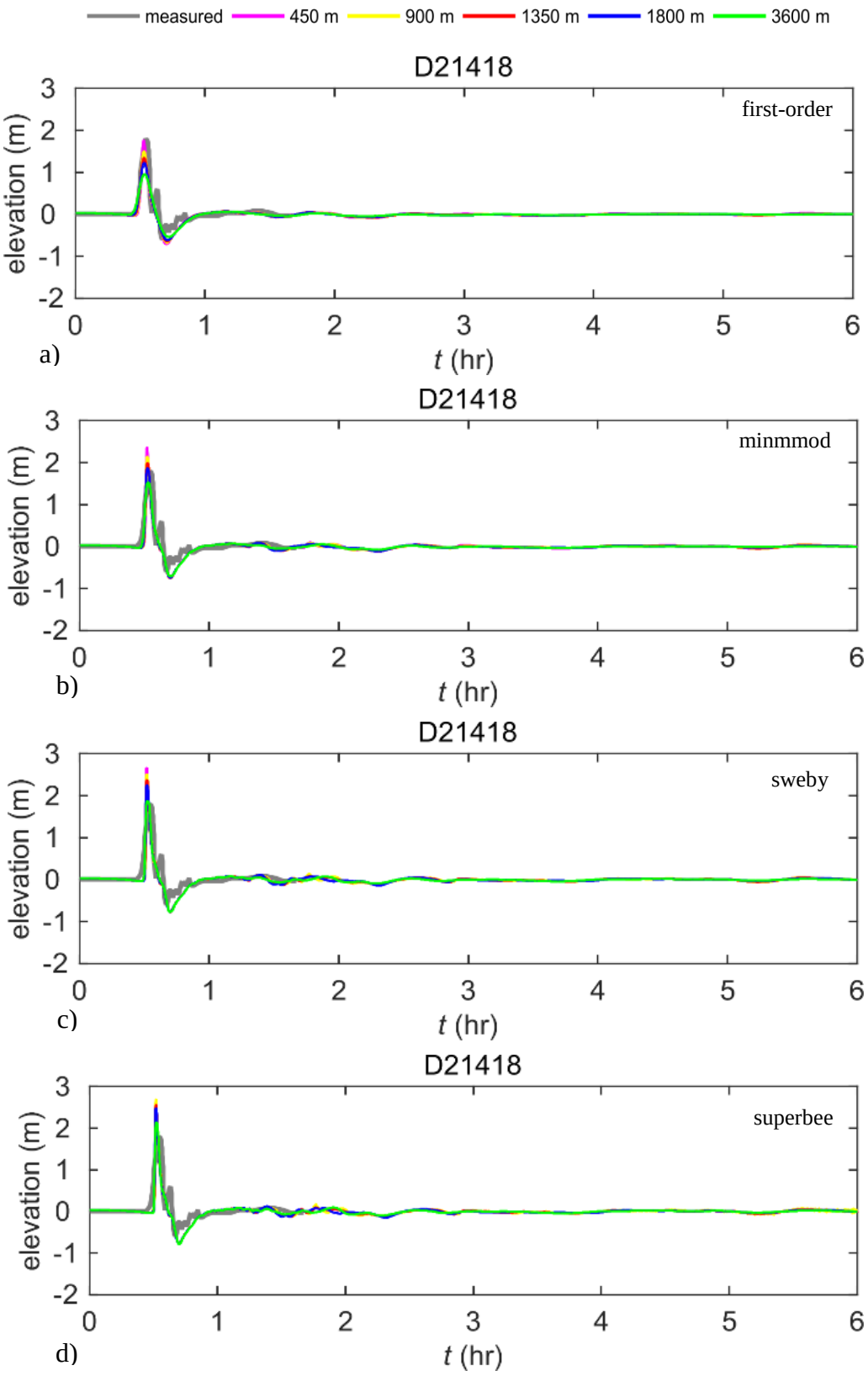


Figure 5.35: The sensitivity analysis results considering different resolutions and slope limiters for 'source 2' at gauge D21418, a) first-order accuracy b) minmod limiter c) sweby limiter d) superbee limiter.

Δx	minmod (E %)	Sweby (E %)	Superbee (E %)	First-order (E %)	Observed
450 m	4.41 m (+10.3 %)	5.54 m (+38.5 %)	unstable	2.97 m (-25.6 %)	4 m
900 m	3.84 m (-4 %)	4.79 m (+19.8 %)	5.51 m (+37.8 %)	2.37 m (-40.8 %)	
1350 m	3.22 m (-19.5 %)	4 m (0 %)	4.5 m (+12.5 %)	2 m (-50 %)	
1800 m	2.83 m (-29.3 %)	3.54 m (-11.5 %)	3.93 m (-1.8 %)	1.75 m (-56.3 %)	
3600 m	1.93 m (-51.8 %)	2.33 m (-41.8 %)	2.48 m (-38 %)	1.23 m (-69.3 %)	

(a) GPS buoy 807

Δx	minmod (E %)	Sweby (E %)	Superbee (E %)	First-order (E %)	Observed
450 m	5.93 m (-5.9 %)	6.81 m (+8.2 %)	unstable	4.90 m (-22.2 %)	6.30 m
900 m	5.59 m (-11.3 %)	6.31 m (+0.2 %)	6.73 m (+6.8 %)	4.14 m (-34.4 %)	
1350 m	5.25 m (-16.7 %)	5.83 m (-7.5 %)	6.47 m (+2.6 %)	3.63 m (-42.3 %)	
1800 m	4.73 m (-24.9 %)	5.62 m (-10.8 %)	6.15 m (-2.4 %)	3.30 m (-47.6 %)	
3600 m	4.01 m (-36.3 %)	4.34 m (-31.2 %)	4.63 m (-26.5 %)	2.91 m (-53.7 %)	

(b) GPS buoy 804

Δx	minmod (E %)	Sweby (E %)	Superbee (E %)	First-order (E %)	Observed
450 m	6.75 m (+1.1 %)	7.52 m (+12.7 %)	unstable	5.89 m (-11.7 %)	6.67 m
900 m	6.52 m (-2.3 %)	7.42 m (+11.2 %)	7.86 m (+17.7 %)	5.35 m (-19.8 %)	
1350 m	6.27 m (-6.0 %)	6.89 m (+3.2 %)	7.32 m (+9.6 %)	4.88 m (-26.9 %)	
1800 m	5.87 m (-12 %)	6.75 m (+1.2 %)	7.28 m (+9.1%)	4.39 m (-34.2 %)	
3600 m	5.17 m (-22.6 %)	6.15 m (-7.8 %)	6.64 m (-0.5 %)	3.60 m (-46.1 %)	

(c) GPS buoy 802

Δx	minmod (E %)	Sweby (E %)	Superbee (E %)	First-order (E %)	Observed
450 m	6.44 m (+13.3 %)	6.66 m (+17.2 %)	unstable	6.01 m (+5.7 %)	5.68 m
900 m	6.34 m (+11.6 %)	6.74 m (+18.6 %)	6.90 m (+21.5 %)	5.34 m (-6.1 %)	
1350 m	6.19 m (+9 %)	6.60 m (16.1 %)	6.88 m (+21.2 %)	4.77 m (-16 %)	
1800 m	5.84 m (+2.8 %)	6.17 m (+8.6 %)	6.37 m (+12 %)	4.18 m (-26.5 %)	
3600 m	5.07 m (-10.8 %)	5.90 m (+3.8 %)	6.35 m (+11.8 %)	3.05 m (-46.3 %)	

(d) GPS buoy 803

Δx	minmod (E %)	Sweby (E %)	Superbee (E %)	First-order (E %)	Observed
450 m	8.25 m (+42.7 %)	8.80 m (+52.3 %)	unstable	6.83 m (+18.2 %)	5.78 m
900 m	7.84m (+35.7 %)	8.55 m (+48.1 %)	9.02 m (+56.1 %)	5.77 m (-0.1 %)	
1350 m	7.38 m (+27.7 %)	8.24 m (+42.7 %)	8.71 m (+50.8 %)	5.04 m (-12.7 %)	
1800 m	6.96 m (+20.5 %)	7.97 m (+38 %)	8.68 m (+50.2 %)	4.59 m (-20.5 %)	
3600 m	6.78 m (+17.4 %)	8.67 m (+50 %)	9.5 m (+64.4 %)	3.52 m (-39.1 %)	

(e) GPS buoy 801

Δx	minmod (E %)	Sweby (E %)	Superbee (E %)	First-order (E %)	Observed
450 m	1.54 m (-41.1 %)	1.63 m (-37.7 %)	unstable	1.36 m (-48 %)	2.62 m
900 m	1.51 m (-42.4 %)	1.64 m (-37.4 %)	1.70 m (-35.1 %)	1.23 m (-53 %)	
1350 m	1.46 m (-44.4 %)	1.62 m (-38.2 %)	1.65 m (-37.2 %)	1.15 m (-56 %)	
1800 m	1.35 m (-48.3 %)	1.52 m (-42.1 %)	1.61 m (-38.7 %)	1.10 m (-58.1 %)	
3600 m	1.17 m (-55.5 %)	1.31 m (-49.9 %)	1.44 m (-45.2 %)	1.02 m (-61.2 %)	

(f) GPS buoy 806

Table 5.6: Comparison of the observed maximum wave amplitudes with the 'source 2' predictions obtained using different resolutions and slope limiters at (a) GPS buoy 807, (b) GPS buoy 804, (c) GPS buoy 802, (d) GPS buoy 803, (e) GPS buoy 801, (f) GPS buoy 806. The numbers in brackets represent the relative errors of the model predictions compared to the observations.

Δx	minmod (E %)	Sweby (E %)	Superbee (E %)	First-order (E %)
450 m	19.1 %	27.8 %	unstable	21.9 %
900 m	17.9 %	22.5 %	29.2 %	25.7 %
1350 m	20.6 %	18 %	22.3 %	34 %
1800 m	23 %	18.7 %	19 %	40.5 %
3600 m	32.4 %	30.7 %	31.1 %	52.6 %

Table 5.7: The mean absolute error of 6 GPS buoys (807, 804, 802, 803, 801, and 806) with the 'source 2' predictions obtained using different resolutions and slope limiters.

5.5 Potential sources of errors in tsunami simulation

The previous section presents the simulation results for the 2011 Japan tsunami using the current GPU-based tsunami model. Extensive numerical simulations have been carried out to investigate the sensitivity of the model results to different tsunami source/fault models (providing initial conditions for the simulations) and other model settings including the use of different spatial resolutions and slope limiters. The uncertainties associated with these model settings may have important implications for real-world tsunami simulations. These, together with other major uncertainties in the tsunami simulation results, are summarized as follows:

- The tsunami wave formation and propagation are found to be highly sensitive to the initial surface water displacement (initial conditions) and so uncertainty arising from different source models can significantly affect the tsunami predictions.
- The data information such as bathymetry/topography which can be obtained from different sources might not be that accurate to represent the reality properly. Furthermore, interpolation techniques used to calculate the data for a specific resolution based on available resolutions would be an estimation rather than exact representation of the desired resolution.
- The observed data which is usually used for model validation might have inaccuracies due to different factors e. g., instrument error.
- It is usually a common practice to compare the simulation results with other available results in the literature to evaluate the model. But the other models might have executed the simulation under a different condition e.g., bathymetry/topography, friction, tsunami source model, and etc. It should be added, the simulations were carried out with and without bottom friction, and both cases reproduced similar results. Therefore, the bottom friction did not affect the simulations of wave propagation in deep-ocean as

expected. However, it can make a significant difference for inundation studies (Arcos and LeVeque, 2015).

- In order to take into account the earth curvature, spherical coordinates is generally used to solve the hydrodynamic equations for the large-scale simulations, which is essential for the continental-scale tsunami simulations. This work focuses more on the near-field tsunami simulations and hence adopts the Cartesian coordinate system, which is assumed to have negligible effects on the simulation results. But further research should be done to evaluate the current results and confirm the assumption. It should be noted that the conversion between the spherical and Cartesian coordinate systems may also produce errors. For example, the locations of subfaults are usually provided in Latitude and Longitude coordinates. In the Cartesian grid based tsunami simulations, the Latitude and Longitude coordinates must be projected to the Cartesian system in a source model to initialize the tsunami waves. The coordinate conversion may slightly distort the 'real' locations of the subfaults, giving a potential source of error/uncertainty in the simulation results.
- The exact positions of gauges where measurements are available for verifying the simulations results are also important. The positions of gauges are recorded in the Lat/Lon system and must be projected to the Cartesian system with small errors. Furthermore, when coarse resolution is used in the simulations and the dislocation of a gauge in the computation domain by just a few cells may mean a few kilometers of mismatch with the 'real' gauge positions. This will affect the result comparison significantly, especially for the gauges closer to the coast in the shallow region where the bathymetry and wave dynamics may vary abruptly.
- Various hydrodynamic models as discussed in the literature may perform slightly differently under the same conditions. While nonlinear shallow water equations can provide an acceptable degree of accuracy, the alternative extended shallow flow models considering numerical dispersion and non-hydrostatic terms (e.g. the Boussinesq models) may be able to produce better results for wave propagation in the relatively deeper water. However, for the nearfield tsunami simulations there should not be major difference between non-linear SWEs and other extended shallow flow models.

In this work, the above errors/uncertainties have been considered and minimized, whenever possible, to provide more reliable simulations.

5.6 The computational time of tsunami waves

Following the validation of the current GPU-based hydrodynamic model, the run time of the model is evaluated by comparing with the conventional simulation on CPUs using sequential codes. For better illustration and comparison of the computational time, the run time concerning various spatial resolutions on different devices are recorded for simulating 1 hour of the tsunami propagation.

The run time and speed up of the GPU model versus CPU model concerning different resolutions are presented in Table 5.8. The run time on a single CPU node is used as the reference for the computational performance comparison. The simulations implemented with OpenMP memory shared parallelization is shown to gain over three times of computational efficiency on a quad-core PC compared with the single-core simulation and the speedup is consistent across different resolutions. Considering the fact that implementation of OpenMP parallelization does not involve major modifications to the existing codes, it provides a straightforward way to achieve high-performance computing on modern PCs that are normally equipped with multiple cores.

In terms of the GPU simulations, the simulations for 1350 m and 1800 m resolutions only require 53 s and 23 s of run time, whereas the same simulations on a single CPU take roughly about 45 min and 20 min respectively to complete. Therefore, for the resolution in the range of 1-2 km which is normally used for the tsunami propagation, the GPU hydrodynamic model can boost the simulation time by more than 50 times compared to the same model running on a single CPU.

Δx	Number of cells	CPU single-core	CPU quad-core	GPU GTX 560Ti	GPU Tesla M2075
450 m	12,150,000	103,550 s (~1 ×)	30,456 s (~3.4×)	memory failure	1538 s (~67.3×)
900 m	303,7500	9734.6 s (~1×)	2949.9 s (~3.3×)	192.6 s (~50.5×)	176.6 s (~55.1×)
1350 m	1,350,000	2744.9 s (~1×)	857.8 s (~3.2×)	58 s (~47.3×)	52.8 s (~52×)
1800 m	759,000	1149.6 s (~1×)	359.2 s (~3.2×)	25.4 s (~45.3×)	23 s (~50×)
3600 m	189,750	138.2 s (~1×)	44.6 s (~3.1×)	3.6 s (~38.4×)	3 s (~46×)

Table 5.8: Computation time for tsunami simulations on different hardware devices. The numbers inside the brackets indicate the model speed up in comparison to the corresponding sequential simulation on a single CPU core. The recorded run times are for the simulation of 1 hour of the physical tsunami event.

As also shown are the simulations carried out for two different types of GPUs, including the regular desktop GPU GTX and the more advanced Tesla GPU fitted in a server. The Tesla GPU can only deliver slightly better performance due to the greater number of cores compared to the GPU GTX. Therefore, the inexpensive regular GPU running on a personal computer is capable of delivering simulations with good performance, compared to the more expensive GPU running on a server. However, the desktop GPU could not handle the larger simulations (e.g. 450 m) as presented here due to the memory limitation (1GB).

As discussed in Chapter 4, for the larger size of the problem, the performance of GPUs can increase and converge to a certain level. In this regard, for the fine-resolution simulations involving 12.15 million cells, the GPU model only needs 25 minutes of runtime while the CPU model takes about 28 hours, giving a speedup of 67 times.

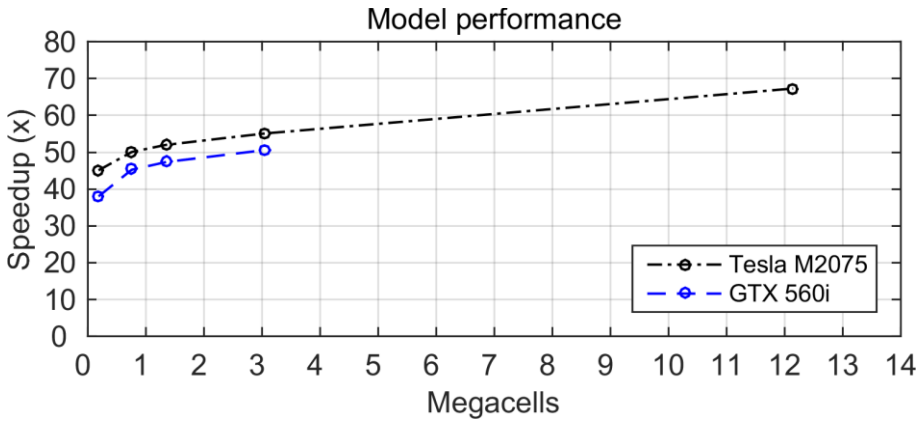


Figure 5.36: The model speed up with different number of cells in comparison to the corresponding sequential simulation on a single CPU, for two different GPUs.

The GPU performance for the two GPUs versus the number of cells used are illustrated in Figure 5.36 which shows a speedup in the range of 40 -70 times for the simulations carried out for this test case.

5.7 Summary

This chapter showed the capability of the GPU-based hydrodynamic tsunami model for the real-world test case of the 2011 Japan tsunami. Two different tsunami source models have been used, one based on the tsunami waveform inversion and another one derived using deep ocean tsunameter measurements. The vertical seafloor deformation computed by the Okada model from the instantaneous rupture during the earthquake is used as the initial conditions to drive the hydrodynamic model.

The model is validated by comparing the simulation results with the observations or other alternative numerical results that are available in the literature. The measurements used for

comparison are available from four different types of gauges including 15 stations from shallow regions along the east coast of Japan extended to the deep ocean at various locations.

The simulation results based on two source models are first evaluated by visually comparing with the measured time series of the entire tsunami wave and then the model accuracy is measured quantitatively for the first leading wave. On average, the simulation results for the waveform including the maximum amplitude of the wave and phase followed by the trailing wave show a good agreement in comparison to the recorded data or alternative solutions using different models. However, the tsunami source derived by the inversion from deep ocean tsunameters ('source 2') provides more favourable accuracy than the source model derived from the waveform inversion under a similar condition ('source 1'). Then after the preliminary validation of the hydrodynamic model carried out at 1350 m resolution with the minmod slope limiter, an intensive sensitivity analysis is attempted for simulations on different spatial resolutions and different slope limiters. The simulations are carried out for a range of resolutions between 450 m and 3600 m using models implemented with minmod, Sweby and Superbee limiters and also first-order accurate scheme. It is concluded that the simulations can provide acceptable results for coarse grid resolutions up to 2 km although the waveform damping becomes evident for simulations with coarser resolutions. The resolutions in the range of 1-2 km seem to provide good results with a proper limiter compared to the finer resolution, and can save computational time. In this regard, the Sweby limiter is found to be the optimized limiter for coarse resolution simulations and to recover the wave peaks better than the more dissipative minmod limiter, while the numerical solution remains stable in contrast to the Superbee limiter. As expected, the first-order accurate scheme shows the worst results for tsunami modelling due to excessive numerical dissipation, especially at coarser resolutions.

Finally as an important aim of the project is to improve the computational efficiency for large-scale tsunami modelling, the required run time for predicting tsunami wave propagation is recorded and compared across different hardware devices with the sequential and parallel versions of the codes. It is shown that one hour of tsunami wave propagation can be predicted in less than a minute by the application of GPU parallel technology, compared to the 45 minutes of run time required by the sequential version of same model running on a single CPU core. GPU simulations are carried out with different spatial resolutions on a regular desktop GPU and a more professional GPU fitted in a server. In all cases, a single GPU can speed up the simulation by at least one order of magnitude. With coarse resolutions roughly in the range of 1-2 km, the computational time may be reduced by more than 50 times. For finer resolution, due to more effective use of GPU cores and the balancing of the overhead communication

between CPU-GPU, the speedup may be increased further and reach up to 70 times. Most of the simulations can be carried out with a cheap regular GPU running on a PC and the performance is comparable to an expensive GPU running on a server.

The adaptive mesh refinement technique on CPU and GPU for tsunami propagation will be introduced in the next chapter.

Chapter 6 Adaptive Grid Technologies for Tsunami Modelling

This chapter introduces a simplified dynamic adaptive mesh refinement (AMR) technique for tsunami simulations. The uniform grid based SWE tsunami model introduced in Chapter 3 is implemented herein on the adopted adaptive structured grid system. Part of the materials in this chapter have already been published in Liang *et al.* (2015a). Attempt has also been made to implement the adaptive grid based model on GPUs for parallel computing, with a focus on static adaptive grids herein. Two test cases are then considered to validate the new components of the tsunami model, followed by a real-world tsunami case.

6.1 Introduction

Tsunami propagation can occur over a large area in the ocean and uniform fine mesh over the whole domain may cause unaffordable computational cost and memory, subsequently limiting the operational forecasting capability of tsunami models. So, it is desirable to refine the grid locally near to the wave fronts and other necessary areas instead of the entire domain. This can be achieved through AMR techniques, which have been a continuous topic of research for over three decades (Berger and Oliger, 1984; Yiu *et al.*, 1996; Pearson *et al.*, 1997; Lee *et al.*, 2011).

Although AMR can be achieved using triangular unstructured grids or other types of grids, this work will focus on Cartesian grids. Two grid adaptation approaches have been widely used in recent years to facilitate AMR on Cartesian grids, which are known as block adaptation (George and LeVeque, 2008; Cowles, 2013; Arcos and LeVeque, 2015) and hierarchical grid adaptation (Popinet, 2003; Liang and Borthwick, 2009; Popinet, 2011). Block grid adaptation generates a group of grids of different resolution to achieve local refinement and grid adaption. This approach has been successfully applied to tsunami simulations by Berger *et al.* (2011) and Watanabe *et al.* (2012), among others.

On the other hand, the hierarchical grid approach mainly uses a quadtree data structure to store grid information, including grid subdivision levels and neighbours. The grid and flow information is updated by searching the data tree during grid adaptation. Quadtree method performs AMR at the individual cell level rather than manipulating a block of grid cells as in the block approach. This provides greater flexibility in refining those areas of interests and the approach has been used by many researchers for shallow flow modelling (Yiu *et al.*, 1996; Popinet, 2003; Liang *et al.*, 2004). Particularly, Popinet (2011) and Popinet (2012) used dynamically quadtree adaptive grids for tsunami modelling.

Although the quadtree grid approach has been proved to be an effective way to provide AMR simulations, it requires large storage for storing grid information and substantial additional computational overhead for grid manipulation. Liang (2012) developed a simplified structured adaptive grid system that uses simple algebraic relationships to specify neighbouring cells, and thus removes the storage requirement of the quadtree grid approach. To generate an initial grid, it only involves allocation of subdivision levels to the coarse background cells and ensures no neighbouring cells are more than two times bigger or smaller than the cell under consideration. Grid adaptation can be easily achieved by changing the subdivision level of a background cell to provide local refinement according to specific criteria. On such a non-uniform grid, the governing equations are solved in the same way as on a uniform grid after applying simple linear interpolation procedure. More details of this simplified grid system can be found in Liang (2012). The model is applied and adapted in this work for tsunami simulations.

6.2 Dynamic adaptive grid

Herein the aforementioned structured adaptive grid system is briefly reviewed and demonstrated for effective tsunami simulations.

6.2.1 *Grid generation and neighbour identification*

A structured but non-uniform grid can be generated in three steps:

1. The computational domain is first fitted into a rectangle and discretised using a uniform coarse grid of subdivision level 0, i.e. “background grid”. For instance, Figure 6.1(a) shows a background grid with 1 × 3 coarse cells, all having zero subdivision level.
2. On the background grid, all of the cells are checked and refined by allocating a specific subdivision level according to certain criteria, as shown in Figure 6.1(b).
3. The final grid is then generated by imposing a 2 : 1 rule. This is to simply ensure that an arbitrary cell does not have a neighbour that is more than two times smaller or bigger. For example in Figure 6.1(b), cell (i, j) does not meet the 2:1 rule because it is four times bigger than its eastern neighbour cell $(i+1, j)$. Therefore, cell (i, j) is regularized by one level of refinement from zero to one, as shown in Figure 6.1(c).

After grid generation, each cell on the non-uniform grid is identified by four indexes (i, j, is, js) as illustrated in Figure 6.1(d). Considering cell (i, j), the sub-grid is indexed by is and js from 1 to Ms. Ms is the number of sub-cells inside (i, j) in the x or y-direction. $Ms = 2^{lev}$, and lev is the subdivision level of cell (i, j). With the size of a background cell (i, j) set to be Δx and Δy, the dimensions of its sub-cells can be calculated by $\Delta x_s = \Delta x/2^{lev}$ and $\Delta y_s = \Delta y/2^{lev}$, respectively.

On such a non-uniform grid, no data structure is needed to store the neighbour information. Neighbours of a cell can be directly determined by simple algebraic relationships. For those cells with the same subdivision levels, the neighbouring cells can be simply obtained as on a uniform grid. When the neighbouring cells have different subdivision levels, for example the sub-cells (i, j, 1, 1) and (i, j, 1, 2) in Figure 6.1(d) have coarser western neighbours, their neighbour can be simply presented as (i-1, j, is_N, js_N), with is_N and js_N are specified by the following simply algebraic expressions:

$$is_N = 2^{lev_N},\ js_N = Ceiling\ (js/2) \tag{6.1}$$

where *Ceiling* (js/2) represents the smallest integer not less than js/2. When the neighbours are in higher division levels, for example sub-cell (i, j, 2, 1) Figure 6.1(d), its two finer eastern neighbours can be indexed by (i+1, j, is_N, js_N), with

$$is_N = 1,\ js_N = js \times 2 - 1 \text{ or } js_N = js \times 2 \tag{6.2}$$

The other neighbours can be specified in a similar way. Readers can refer to Liang (2012) for more details.

6.2.2 *Grid adaptation indicator*

The first step to achieve grid adaptation is to define the adaptation criterion. In this regard, the averaged water surface is employed to control grid adaptation and capture the key features of tsunami hydrodynamics, e.g. the propagating wave fronts. The averaged water surface gradient at cell ic is defined as:

$$G_{ic} = \sqrt{\left(\frac{\partial \eta}{\partial x}\right)_{ic}^2 + \left(\frac{\partial \eta}{\partial y}\right)_{ic}^2} \tag{6.3}$$

based on which the final non-dimensional grid adaptation indicator is defined as:

$$\Theta_{ic} = \frac{G_{ic}}{P\mathrm{q}(\mathbf{G})} \tag{6.4}$$

where **G** is a vector containing water surface gradients for all of the sub-cells inside the computational domain, $P_q(\mathbf{G})$ returns the q^{th} percentile of **G**, $q = 1 - s_a$, and s_a is a user defined coefficient indicating the sensitivity of the adaptation procedure. For example, $s_a = 0.2$ indicates that 20% of the flow cells (sub-cells) may be subject to grid refinement. For grid coarsening, a critical value of Θ_{coarse} must also be prescribed. The grid refinement and coarsening procedures are explained further in the following sub-section.

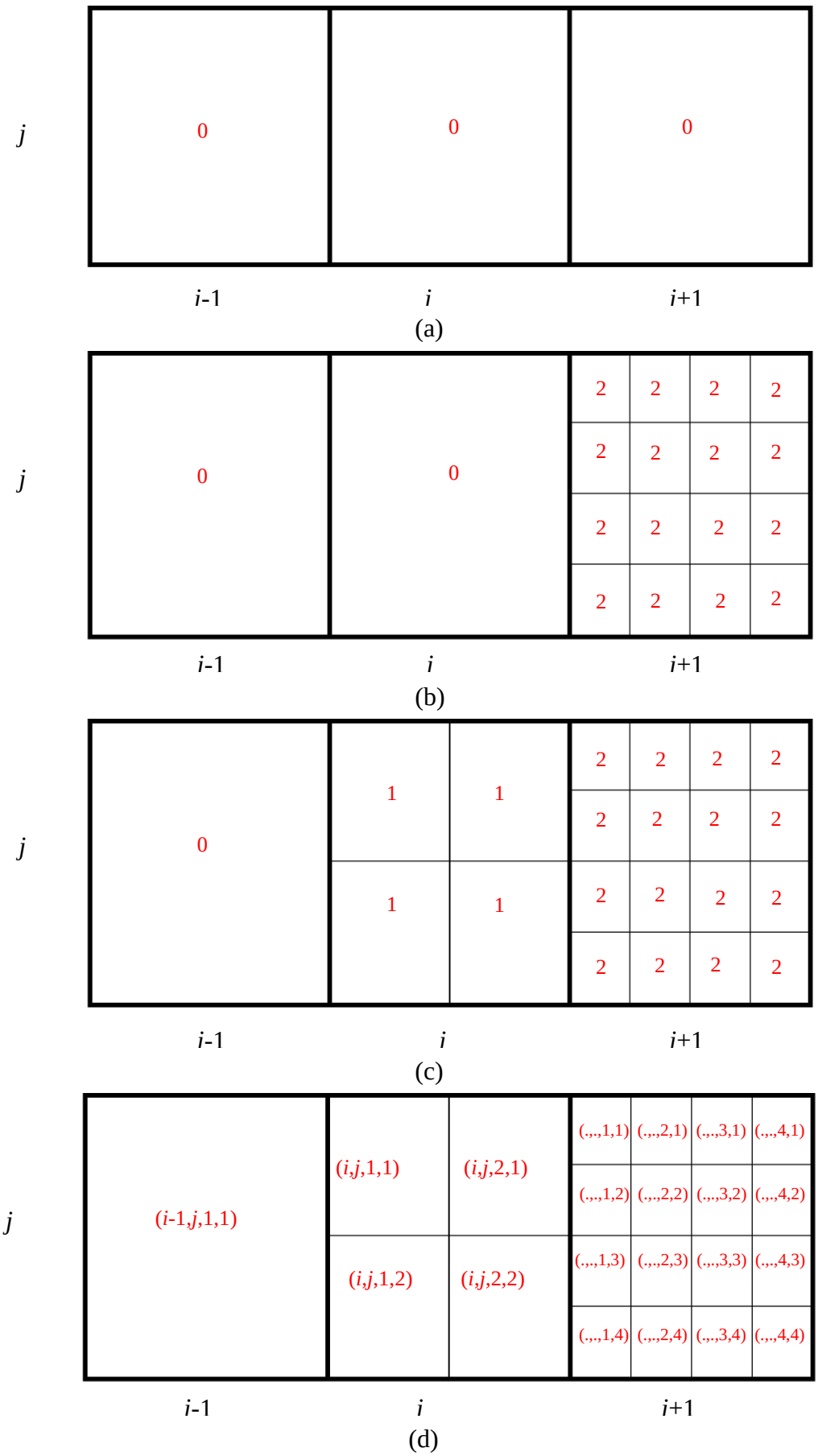


Figure 6.1 Grid generation: (a) background grid; (b) irregular grid; (c) regularised grid; (d) cell indexes for the regularised grid.

6.2.3 *Dynamic grid adaptation*

The grid refinement and coarsening procedures are implemented by the following two steps:

1- In a background cell (i, j) all of the subcells are checked, if any subcell has the value of the adaption indicator Θ_{ic} greater than 1 and the subdivision level of (i, j) is less than the specified maximum value, cell (i, j) will be marked for further subdivision and its subdivision level will be increased to lev + 1. After marking a cell for subdivision, all its eight neighbors are also checked and may be also subdivided in order not to violate the 2 : 1 rule. Any cells representing wet-dry interfaces will be marked for subdivision and ensured maximum grid refinement to capture the wet-dry interfaces.

2- For grid coarsening, if the values of Θ_{ic} of all the sub-cells in (i, j) are lower than Θ_{coarse} and its subdivision level is greater than 0, cell (i, j) will be coarsened by decreasing its subdivision level, i.e. changing lev to lev-1, followed by further checking to ensure the 2:1 rule.

6.2.4 *Conservative flux calculation and flow information on newly created cells*

On an aforementioned simplified structured adaptive grid, the governing shallow water equations are solved using the 1st/2nd-order finite volume Godunov-type scheme as introduced in Chapter 3. The solution procedure is similar to that on a uniform Cartesian grid after applying simple linear interpolation schemes to a limited number of non-uniform grid configurations when the cell under consideration has neighbours of different size (Liang, 2012). A typical grid configuration is shown in Figure 6.2 where cell ic is adjacent to two refined eastern neighbours. To update the flow variable in cell ic, the fluxes through all four of its edges must be estimated in a proper way. The fluxes leaving the eastern interface $\mathbf{F}_E$, must be considered with special care to maintain the conservation property of the finite volume scheme and may be obtained from:

$$\mathbf{F}_E = (\mathbf{F}_{E1} + \mathbf{F}_{E2})/2 \tag{6.5}$$

To calculate the flux vector $\mathbf{F}_{E1}$, flow information is required in cell ic', its eastern neighbor 'e' and location 'w'. The unknown information at w is interpolated from those nearest neighbors using a bilinear interpolation method. The flux vector $\mathbf{F}_{E2}$ may be estimated in the same way. During grid refining and coarsening, new cells of different size will be created. Flow and other relevant information must be obtained for these new cells to facilitate flow calculations and the approach suggested by Liang *et al.* (2015b) is implemented, which resolves the contradiction between the C-property (conservation property) and mass conservation, by locally bed

modifying when reconstructing flow information using linear interpolation during grid adaptation. The capability of the aforementioned approach in simultaneously preserving the water surface conservation (C-property) and mass conservation on adaptive grids is proven mathematically. Moreover, Liang *et al.* (2015b) through different test cases on adaptive grids showed that their technique perform better in terms of numerical robustness and accuracy compared to the conventional approaches that only preserve the C-property or the mass conservation e.g. (Liang and Borthwick, 2009; Popinet, 2011).

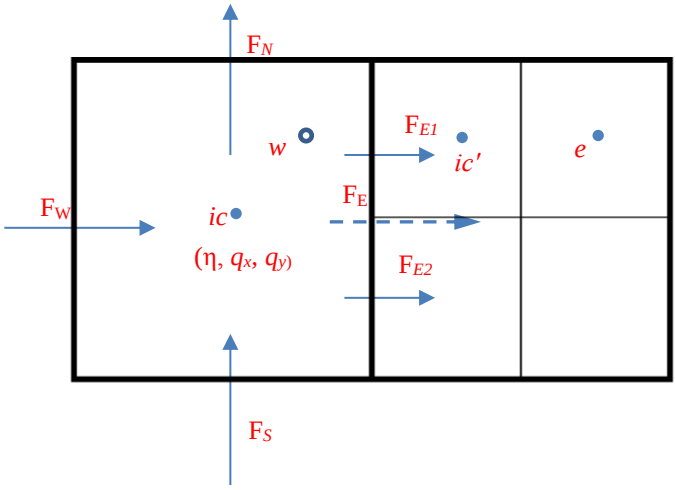


Figure 6.2 Conservative flux computation on a non-uniform grid configuration.

In order to ensure an accurate description of the domain, the bathymetric/topographic values and friction parameters are interpolated directly from the original datasets that are generally available on raster uniform grids at different resolutions covering different parts of the computational domain. In order to reduce the burden on computer memory, the multiple files storing bathymetry data and friction values are stored offline in the local drive and read in during grid adaptation (Popinet, 2011).

6.2.5 *Test case – reconsideration of the Monai tsunami benchmark test*

This test case was used in Chapter 4 to validate the current tsunami model on uniform grids and is reconsidered herein to demonstrate the improved performance of the dynamic adaptive grid based model. Detailed information about this test case can be found in Chapter 4. Particularly related to the current dynamic adaptive grid based simulation, the sensitivity and coarsening parameters are set to be s_a = 0.2 and Θ_{coarse} = 0.7, respectively. The computational domain is covered by a background grid with 98 × 61 cells at a resolution of 0.056 m. Bathymetric data are available at a uniform resolution of 0.014 m over the whole domain, allowing up to 2 levels of refinement/coarsening during the simulation.

Figure 6.3 illustrates the evolution of the free surface elevation as well as the corresponding adaptive meshes. The predicted propagating tsunami wave and run-up appear to be identical to those predicted on the uniform grid as presented in Chapter 4. This is confirmed by the comparison with the laboratory measurements available in the three gauges G5, G7 and G9, as shown in Figure 6.4.

In Figure 6.3, it can be seen that the complicated evolving wave patterns and wave fronts are clearly captured by adaptive meshes. These results are similar to those reported by Popinet (2011) using adaptive quadtree grids. Table 6.1 shows the RMSE of water surface elevation calculated against the laboratory measurements at the three gauges, in comparison with the prediction on the high-resolution (0.014 m) uniform grid conducted in Chapter 4. Close agreement has been achieved, further confirming the aforementioned visualised results. Also presented in Table 6.1, the adaptive grid based model is found to be about three times more efficient than its uniform grid counterpart to obtain results with similar numerical accuracy.

Guage no.	G5	G7	G9	CPU time (s)
RMSE on uniform grid (cm)	0.399	0.358	0.348	561.33 (~1×)
RMSE on adaptive grid (cm)	0.392	0.362	0.344	176.83 (~3.2×)

Table 6.1 Monai tsunami benchmark: RMSE of water surface elevation predicted on uniform and adaptive grids at three gauges. The last column represents the run time respectively required by the 25 s simulations on uniform and adaptive grids.

In next section the GPU architecture for dynamic parallelism is briefly discussed. Then the attempt to implement the static non-uniform grid based tsunami model for GPU computing is introduced, followed by a test case to validation the new model implementation.

6.3 Dynamic adaptive grid on GPU

Earlier CUDA programs had to perform a sequence of kernel launches for a bulk parallel programming model. To achieve better performance, each kernel should contain sufficient parallelism to efficiently use the GPU. The bulk parallel model is appropriate for applications consisting of loops. However, some parallel patterns such as nested parallelism cannot be handled easily (NVIDIA, 2012a). Flat, bulk parallel applications have to consider fine grids and execute unnecessary computations, or use coarser grids which might lose the accuracy of the solution. In this regard CUDA 5.0 and later introduced dynamic parallelism which enables launching kernels from threads running on device, and threads can launch more threads (NVIDIA, 2012a; NVIDIA, 2012d). An application can launch a coarse-grained kernel which in turn launches finer-grained kernels to do the computation where necessary.

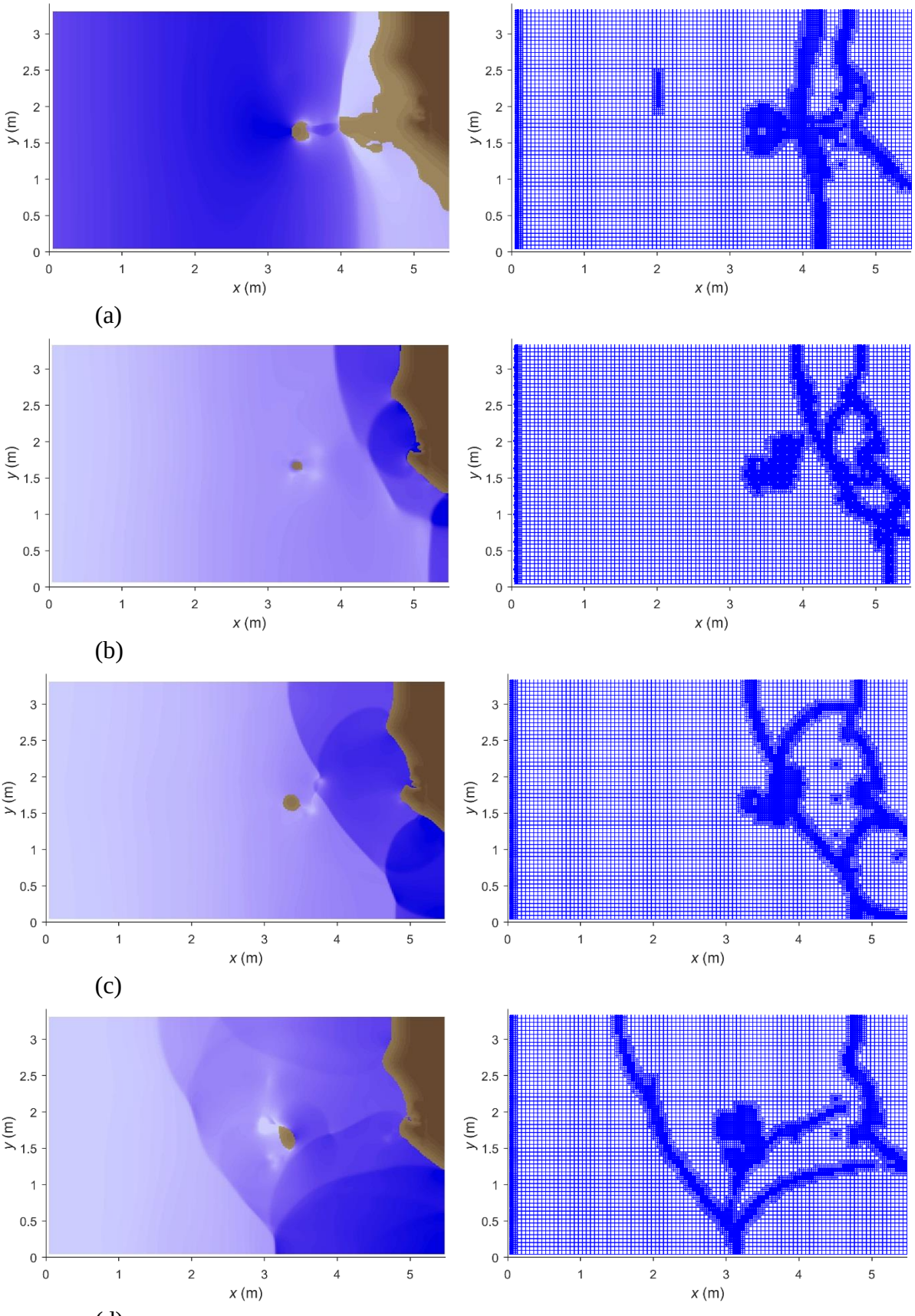


Figure 6.3 Monai tsunami benchmark: free surface elevation evolution (left column) and corresponding adaptive meshes (right column) at (a) t = 14 s; (b) t = 17 s; (c) t = 18 s; (d) t = 20 s.

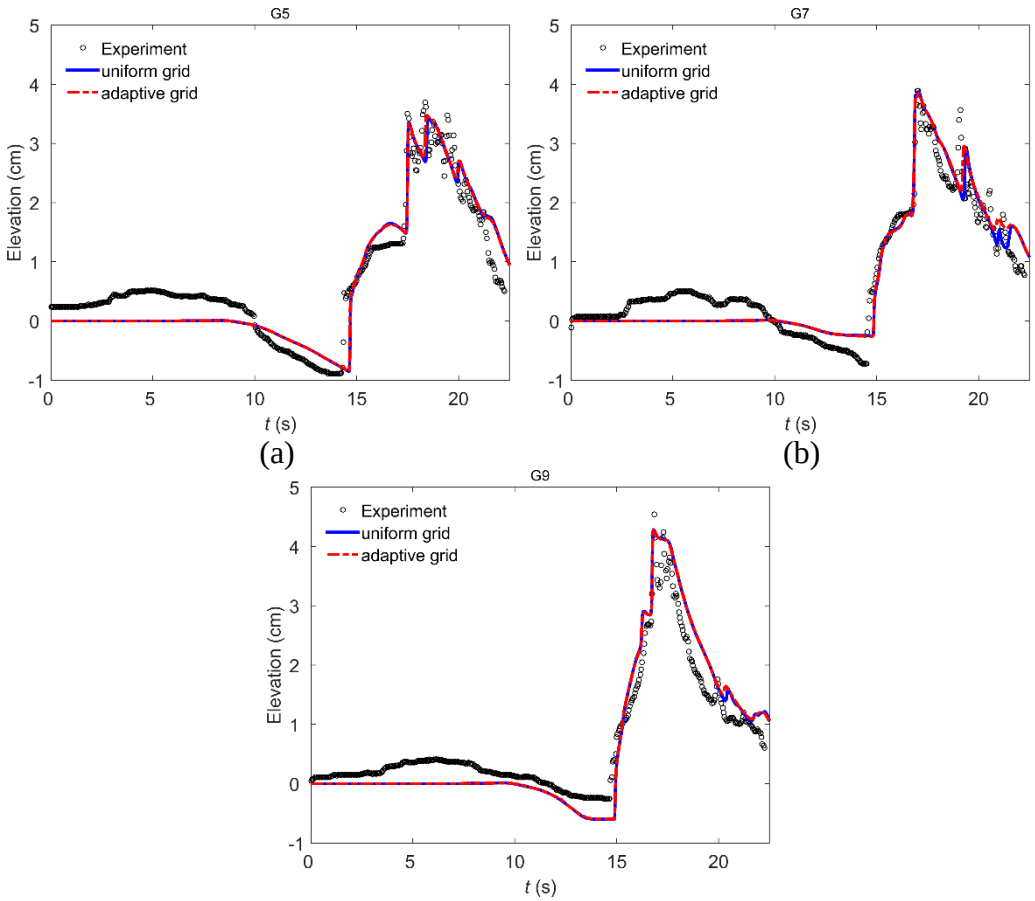


Figure 6.4 Monai tsunami benchmark: comparison of simulation results obtained on uniform and adaptive grids with the experiment measurements at the three gauges: (a) G5; (b) G7; (c) G9.

However, dynamic parallelism is only available on devices of computing capability 3.5 or higher (NVIDIA, 2012a). A good example of these GPUs is the Kepler GK110 released after 2012 which has already been known as the most efficient HPC architecture ever built and also supports dynamic parallelism (NVIDIA, 2012e; NVIDIA, 2012f).

This GPU allows a program to be run directly on the device, as kernels have the capability to independently launch additional workloads as required. With this new feature, nested kernels can be launched without communicating with the CPU as shown in Figure 6.5, considering a single GPU and a single CPU. Illustrated in Figure 6.5 is firstly the Fermi architecture without dynamic parallelism, where the CPU launches each kernel onto the GPU; on the other hand, the Kepler GPU with the new features can launch nested kernels, eliminating the need to communicate with CPU.

However, it should be mentioned that there are alternative algorithms that lift the limitations of the GPU architecture for dynamic parallelism, e.g. the Sierpinski space filling curves (SFC) adopted by a number of researchers (Schive *et al.*, 2012; Ma *et al.*, 2014). The SFC-type

approaches provide a desired advantage to deliver an efficient solution to the memory hierarchy and load balancing challenges in case of using multiple CPUs or GPUs. Because when AMR technique is implemented, the conventional rectangular domain decomposition may not distribute the work load efficiently among all processors and this issue should be solved by redistributing the computational loads (Campbell *et al.*, 2003).

Schreiber (2014) addresses the challenges in high-performance computing related to the dynamic adaptive mesh refinement by implementing an SFC approach on structured triangular grids. The aforementioned wok involves the development of shared and distributed memory on CPUs; but similar concept may be adapted for implementation on GPUs. Xia and Liang (2016) developed a GPU based SPH model for solving the SWEs. In their work, a quad-tree neighbour searching method is implemented to optimize the model performance. In this regard, a z-curve approach, which is similar to an SFC method, was used for memory coalescing to accelerate particle interaction.

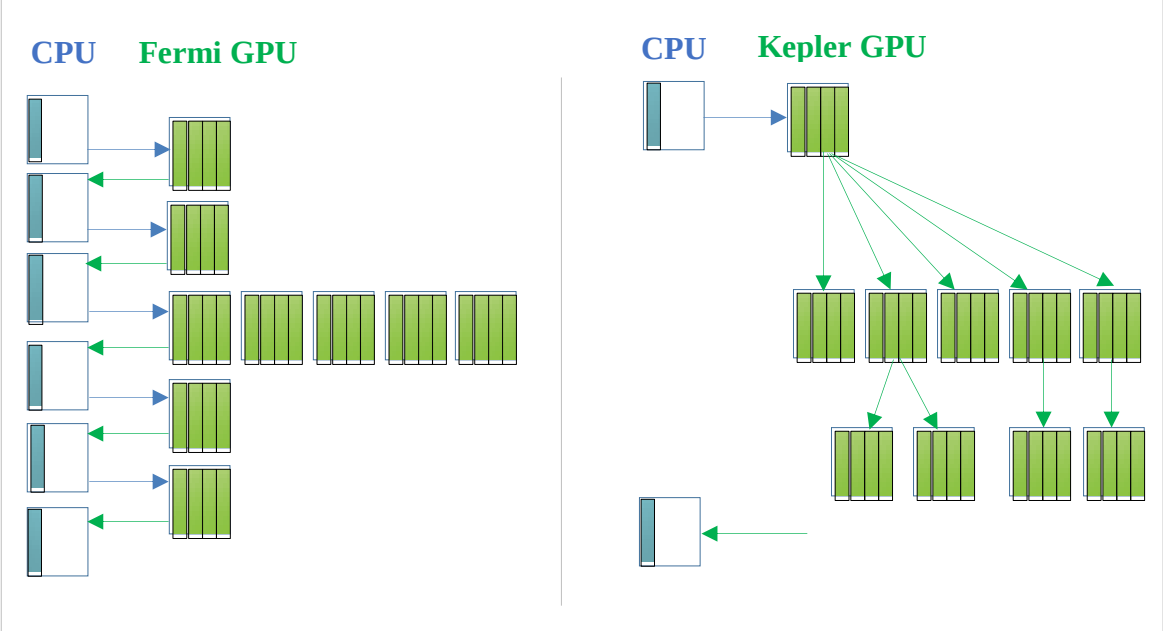


Figure 6.5 Dynamic parallelism on two types of GPU architectures. Right side of the image: the Kepler GPU allows more parallel code in an application to be launched directly by the GPU onto itself. Left side of the image: the Fermi architecture requires CPU intervention (NVIDIA, 2012e).

6.4 Static mesh refinement on GPU

In this section, a similar approach for implementing AMR on CPU as introduced in Section 6.3 is adapted for parallel computing on GPU. Herein, we only consider the static mesh refinement and the locally refined mesh generated to better capture the topographic features and possible complex flow patterns with higher accuracy remaining unchanged throughout the simulations.

In practice, tsunami waves interact intensively with the bed topographic features in the nearshore and coastal regions. Adopting a refined mesh to represent at a higher resolution the bathymetry in the coastal region will potentially provide more accurate prediction of tsunami wave dynamics and run-up. In this regard, embedding the multi-scale bathymetric data with grid generation would provide an effective way to allow the flow solver to capture the complex flow features in those areas of interest at an affordable computational cost. The example of embedding bathymetric data of multiple resolutions near the coast will be presented in the last test case that reproduces the 2011 Japan tsunami.

To demonstrate the generation of a locally refined mesh for tsunami modeling, Figure 6.6 presents a hypothetic computational domain in which the deep sea is extended to the coastal and land areas. While coarse resolution topographic data is available for the whole domain, high-resolution datasets are available for certain parts over the coastal and land areas.

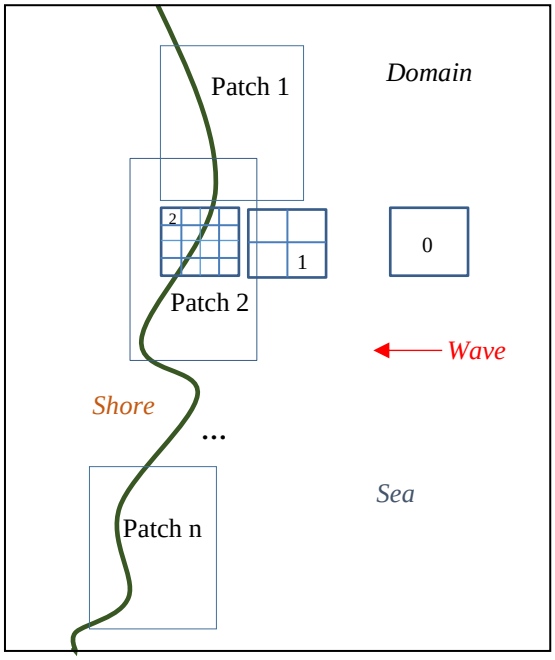


Figure 6.6 A hypothetic computational domain with bathymetric data of multiple resolution: different patches cover the shallow region and land with a finer resolution.

To make the best use of the available high-resolution data and to better represent the bathymetry/topography of the area for a more accurate simulation, the entire domain may be firstly discretized by a coarse background grid with cells having a subdivision level '0'. Depending on the resolution of the data in the patched areas, locally refined meshes will be then generated by simply increasing the subdivision level of the background cells inside the patches. Finally, the entire computational grid will be regularized by reinforcing the 2:1 rule, creating the final grid for the following tsunami simulation. Figure 6.6 illustrates the steps of generating a mesh with two levels of refinement. The cells inside the patches are indexed with a subdivision

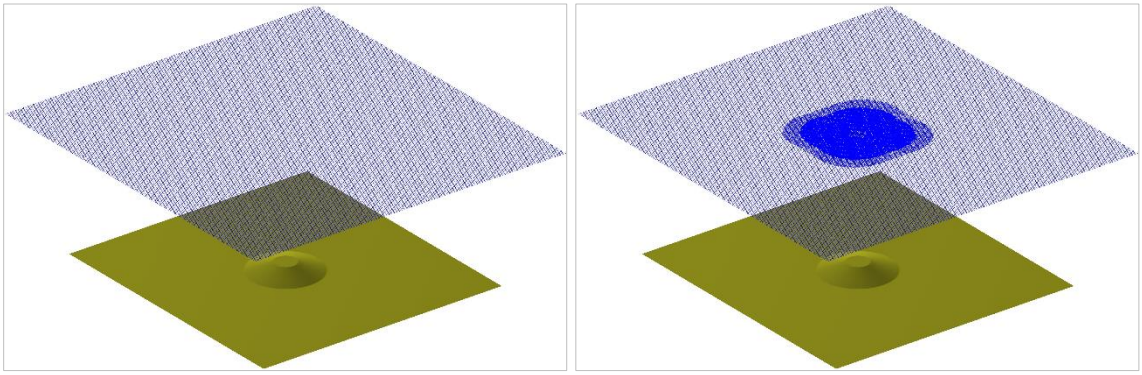
level ‘2’ while the neighboring cells are regularized with a subdivision level ‘1’. Herein, the grid generation is carried out on CPU and the grid data are then transferred to the GPU and used by the relevant kernels to solve the fluid equations.

The new generated grid which is transferred to GPU contains the information for each cell such as index, the location, the bathymetry/topography, and the level of refinement plus the bathymetry/topography of the eight potential neighbouring cells.

6.4.1 *Model validation – reconsideration of conical island tsunami benchmark*

This test case was used in Chapter 4 to validate the current tsunami model on uniform grids in predicting 2D tsunami wave run-up and is reconsidered herein to demonstrate the improved performance of the static adaptive grid based model on GPU.

The computational domain is covered by a background grid with 173 × 162 (28,026) cells at a resolution of 16 cm, on which locally refined mesh with a finest resolution of 4 cm, i.e. two levels of refinement, is generated to represent the island in the middle of the domain, as illustrated in Figure 6.7. The final grid has a total of 48,939 cells in which the SWEs are solved.



(a) (b)

Figure 6.7 Conical island benchmark: (a) background grid; (b) final grid with local refinement around the island.

Figure 6.8 shows the 3D view of the wave propagation and run-up around the island at different times. The predicted propagating wave and run-up appear to be identical to those predicted on the uniform grid as presented in Chapter 4. To further demonstrate the performance of the GPU model on the static adaptive grid, the measured and simulated water elevation at the five specific gauges are shown in Figure 6.9, compared with the results obtained on uniform grids with a fine resolution of 4 cm. Overall, the predicted results are in a good agreement with the measurements. The leading wave heights and the arrival times are well predicted at most gauges. This shows the key features of the propagating waves can be reproduced by the GPU model on non-uniform grids.

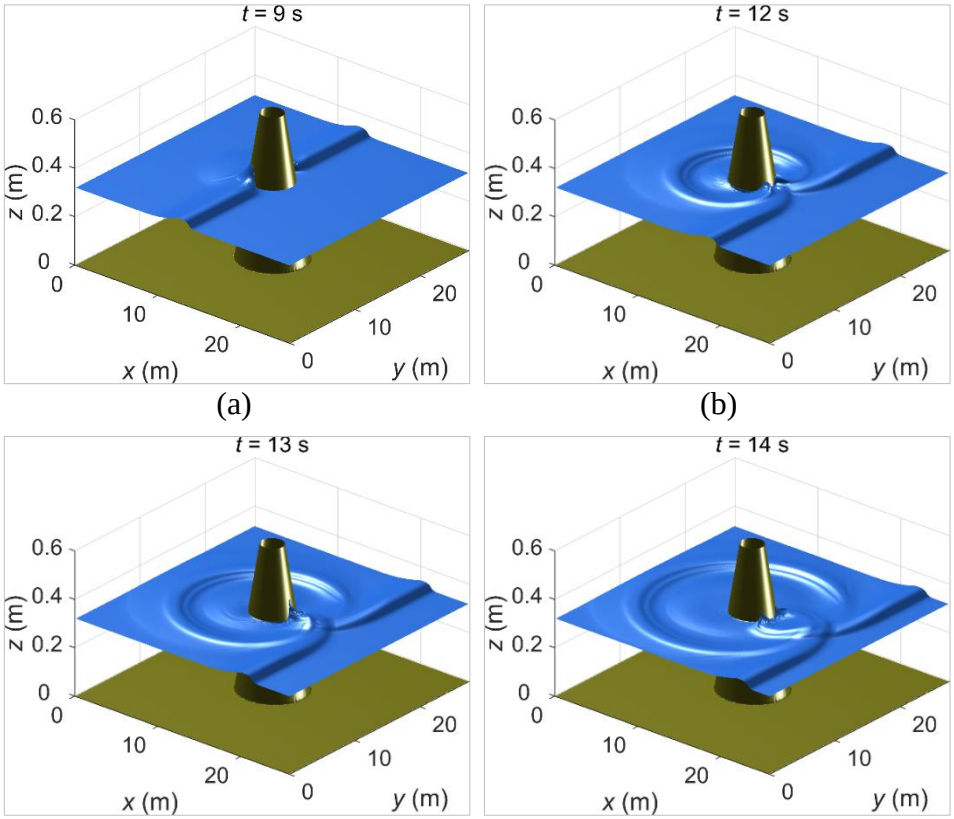


Figure 6.8 Conical island benchmark: 3D view of the simulated wave run-up over the conical island predicted by the GPU non-uniform grid model at different output times (a) t = 9 s; (b) t = 12 s; (c) t = 13 s; (d) t = 14 s.

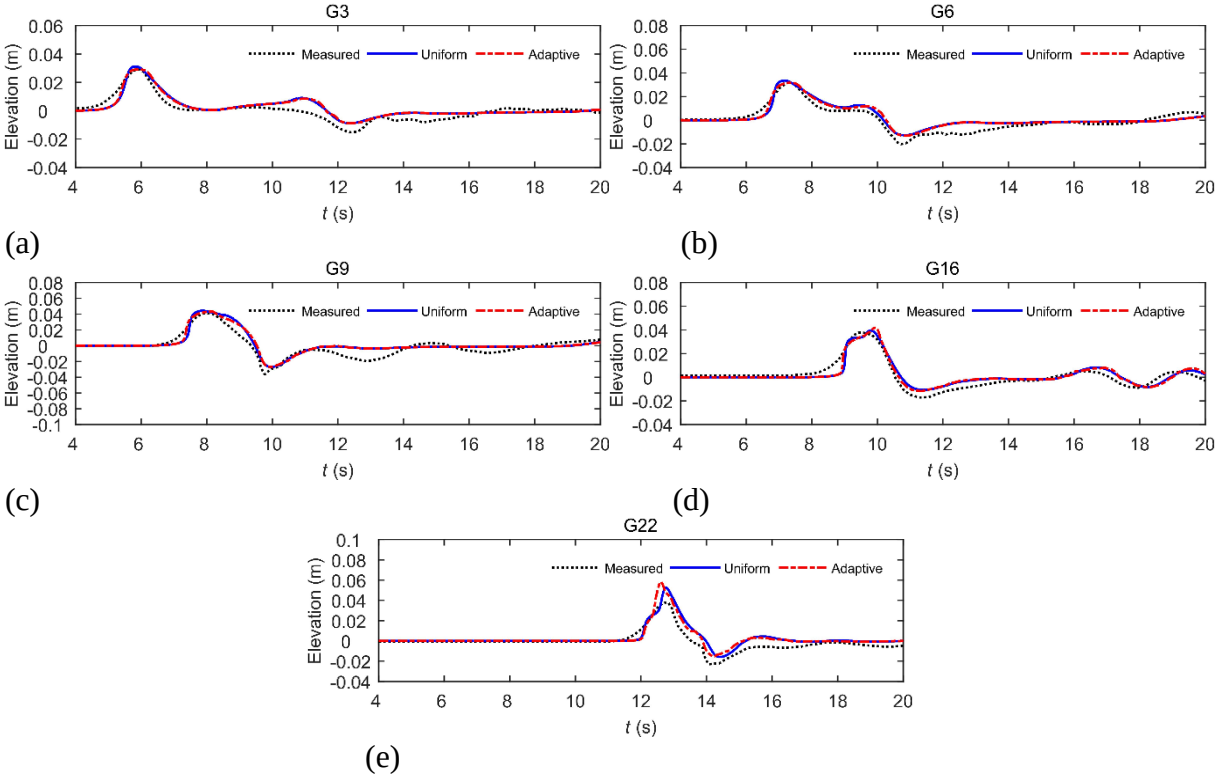


Figure 6.9 Conical island benchmark: comparison between the predicted water surface elevation and measurements at five gauges (a) G3; (b) G6; (c) G9; (d) G16; (e) G22.

The current numerical solutions also compared well with those results reported by other researchers who used alternative SWE models and grid systems (Hubbard and Dodd, 2002; Nikolos and Delis, 2009; Hou *et al.*, 2013).

The RMSEs calculated against the measurements at five gauges are presented in Table 6.2, which further confirms that the locally refined model can predict complex tsunami wave dynamics at numerical accuracy similar to the high-resolution uniform grid. Apparently, much fewer cells are involved in the computation on the non-uniform grid, compared to the high-resolution uniform grid (48,939 vs 447,120), leading to improved computational efficiency. The computational time for the 20 s of simulation required by the static adaptive grid based model is compared with that required by the uniform grid based model in Table 6.3. The GPU non-uniform grid model is found to be over three times more efficient than its uniform grid counterpart for a simulation with similar resolution.

Gauge no.	G3	G6	G9	G16	G22
RMSE on uniform grid (m)	0.0092	0.0070	0.0126	0.0083	0.0102
RMSE on non-uniform grid (m)	0.009	0.008	0.0139	0.0091	0.0134

Table 6.2 RMSE of water surface elevation on uniform and a non-uniform grid at five gauges.

Grid type	Number of cells	GPU
Uniform	447,120	24.84 s (~1 ×)
Static Adaptive	48,939	6.9 s (~3.6 ×)

Table 6.3 Conical island benchmark: computational time required by the GPU models on uniform and static adaptive grids.

6.4.2 *Model validation – reconsideration of the Japan 2011 tsunami*

The 2011 Japan tsunami was simulated and discussed in details at Chapter 5, which is considered here again to demonstrate the performance of the new static adaptive grid based GPU model. For the static adaptive grid or dynamically adaptive simulations, the background grid consists of cells with a coarse resolution of 1800 m, allowing up to two levels of refinement. The refined meshes are initially generated to cover the east coast of Japan that is overlapped by seven patches with higher resolution (450 m) data, as illustrated in Figure 6.10.

Figure 6.11 presents the wave elevation predicted by the models based on high-resolution uniform grid, dynamically adaptive grid and static adaptive grid at four sample gauges, compared with the measured data.

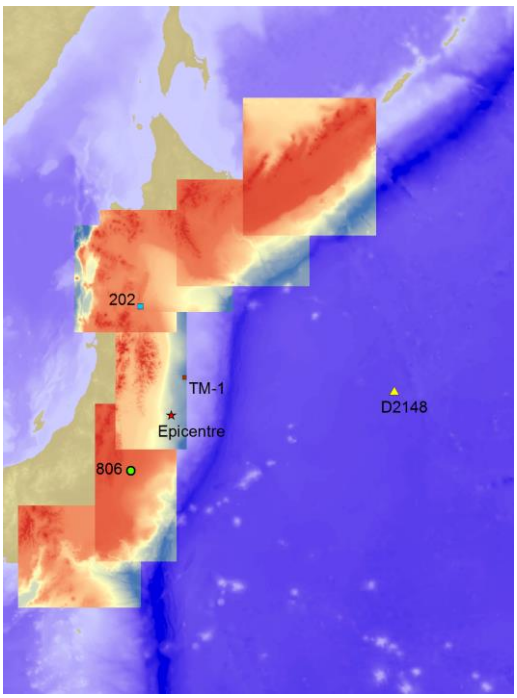


Figure 6.10 Japan 2011 tsunami: background domain with coarse resolution of 1800 m and seven patches along the east coast with finer resolution of 450 m for adaptive grid implementation.

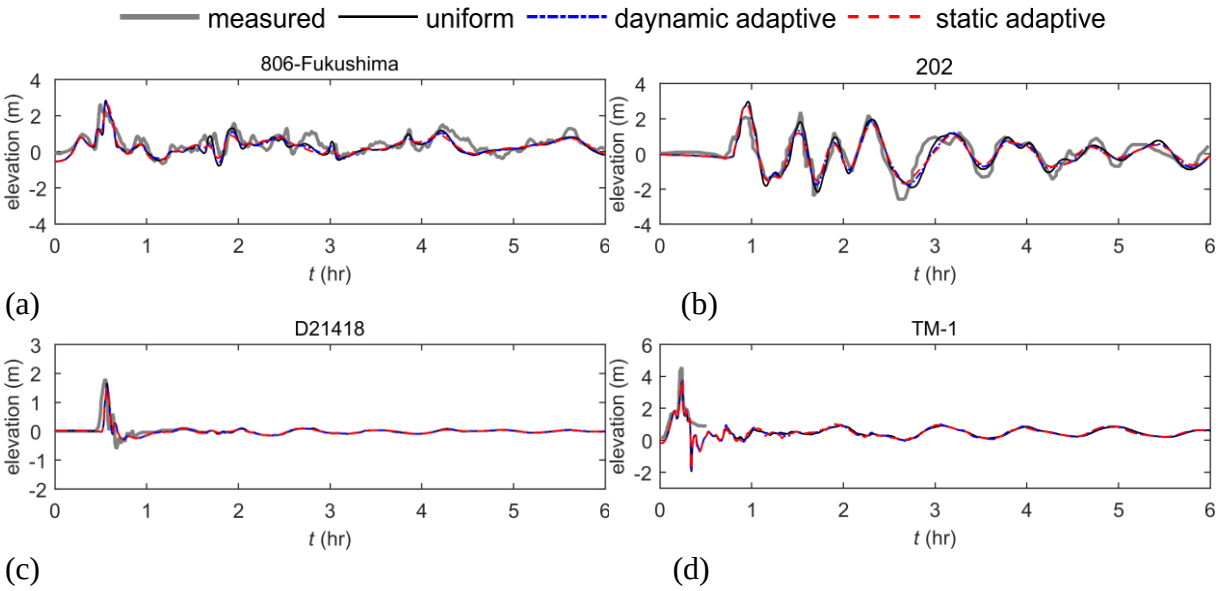


Figure 6.11 Japan 2011 tsunami: simulated and observed wave height at four sample gauges: (a) 806; (b) 202; (c) D21418; (d) TM-1.

Grid type	806	202	TM-1	D2148
Uniform	0.560 (m)	0.484 (m)	0.990 (m)	0.287 (m)
Dynamic adaptive	0.584 (m)	0.471 (m)	0.985 (m)	0.294 (m)
Static adaptive	0.586 (m)	0.445 (m)	0.983 (m)	0.296 (m)

Table 6.4 Japan 2011 tsunami: RMSE of different numerical results calculated against measurements at the four gauges.

The uniform grid simulation is carried out on a 450 m resolution. The dynamic adaptive grid based model only runs on CPU. The models based on different grids perform quite similar for this case and all of the simulation results compare will with the measurements. This is further confirmed by the RMSE values as presented in Table 6.4.

The computation time required by different models for the one-hour simulation is presented in Table 6.5 and Table 6.6, in comparison with that required by the same simulation on uniform grid. The performance gained by implementing adaptive grid with two level of refinement is about three times for both cases compared to the same simulation on uniform grid consistent to the previous test cases.

Grid type	CPU
Uniform	103,550 s (~1 ×)
Dynamic adaptive	36,080 s (~2.87 ×)

Table 6.5 Japan 2011 tsunami: Computation time required by the uniform grid based and dynamically adaptive based models running on CPU (numbers inside the brackets indicate the model speed up in comparison to the corresponding simulation on the uniform grid).

Grid type	Number of cells	GPU
Uniform	12,150,000	1538 s (~1 ×)
Static Adaptive	3,036,402	496.1 s (~3.1 ×)

Table 6.6 Japan 2011 tsunami: Computation time required by the uniform grid based and static adaptive based models running on GPU.

6.5 Summary

In this chapter, certain popular adaptive grid based models for tsunami simulations are reviewed briefly, followed by the introduction of a simplified dynamic adaptive grid based model developed in this work for tsunami simulations on CPU.

Then the capability and limitations of different GPU devices for adaptive parallelism are discussed. As a summary, dynamic parallelism has limitations on a wide range of GPUs unless they are compute capability 3.0 or higher e.g. the Kepler generation of GPUs. Alternative algorithms for adaptive parallelism have also been briefly discussed.

In this work, a static adaptive mesh refinement is developed and has a potential for implementation on a wider range of GPUs as the grids are statically produced at the beginning and remain constant during the simulation. With this technique, embedding multi-scale bathymetries is possible through locally refined meshes to cover those regions of interest near the coast where calculation of the tsunami run-up requires a higher degree of resolution.

The model's numerical accuracy and performance for both of the static adaptive grids and dynamically adaptive grids are evaluated through simulations of two tsunami benchmarks, followed by the reconsideration of 2011 Japan tsunami. The accuracy of the results on adaptive grids is found to be satisfactory in comparison to those obtained on high-resolution uniform grids. By considering only two levels of refinement, the adaptive grid on both CPU and GPU can save computational time by a factor of three while the results maintain an acceptable degree of accuracy.

Chapter 7 Conclusions and Future Work

This chapter provides a brief summary and conclusions which result from this work. The conclusions are first discussed regarding the model validation and performance. Then the summary of the model capability to reproduce the 2011 Japan tsunami is discussed and concluded. Some of these discussions and conclusions can be also found in (Amouzgar *et al.*, 2014; Amouzgar *et al.*, 2016). The adaptive mesh refinement (AMR) techniques which were implemented in this model on CPU and GPU are mentioned in the last section of these conclusions; some parts of the results related to the AMR technique on CPU are published in (Liang *et al.*, 2015a). Finally some possible future directions to continue this research will be mentioned.

7.1 Conclusions

7.1.1 *Hydrodynamic modelling of tsunamis using GPU technology*

In this thesis, a hydrodynamic model based on GPU parallel computing has been presented for tsunami simulations. The model solves the 2D nonlinear SWEs using a MUSCL-Hancock second-order finite-volume Godunov-type scheme incorporated with an HLLC approximate Riemann solver. The model is capable of simulating tsunami propagation and run-up involving advancing bores and moving shorelines in the inundation zone over irregular topographies. The hydrodynamic model is then validated by five different test cases (Chapter 4-Section 4.1). It was shown that the proposed model could reproduce the wave propagation and run-up successfully compared to laboratory measurements or alternative numerical simulations.

The computational performance of the GPU hydrodynamic model was evaluated by comparing to a sequential Fortran model based on the identical numerical scheme that runs on a single CPU core. It was demonstrated that the GPU model could improve the computational efficiency by more than 40 times for all test cases. The performance of the GPU simulations increases as more computational cells are used for the calculations, as the heavier use of GPU balances the CPU-GPU communication overhead. The simulations were executed using two different GPU devices, namely the Tesla M2075 (installed on a server) and Geforce GTX 560 Ti (desktop GPU). It was shown that the performance of the cheaper desktop GPU is comparable to the more advanced server GPU.

The performance of the sequential code was also improved by implementing shared memory parallelization using OpenMP to take the advantage of multi-core CPUs which are available on

modern desktop and laptop computers. Full utilisation of quad-core CPUs on a standard PC could speed up the simulation more than three times (i.e. the model performance scaled-up near linearly) compared to the single core CPU, as was expected.

7.1.2 *Simulation of the 2011 Japan tsunami*

The proposed model in this work was used to reproduce the 2011 Japan tsunami to show the capability of the model for real world simulations. The model could simulate the tsunami wave propagation in offshore and nearshore along the east coast of Japan, and compared well with the observations or other reported results from the literature. Two different source models were adopted, one based on tsunami waveform inversion and an alternative one derived using deep ocean tsunameters; the fault parameters from these models are taken from Fujii *et al.* (2011) and Wei *et al.* (2013) respectively (Chapter 5-Section 5.3). After model validation with a visual check of the wave time-series at 15 gauges, it was concluded that, overall, both source models could initiate a tsunami wave very similar to the real event. It is also concluded that the first wave amplitude and phase are more sensitive to the source models compared to the trailing waves. This is because the trailing waves, especially in the shallower region, are more affected by the local bathymetry and wave shoaling, refraction and reflections. For better evaluation of the source models, the lead wave amplitude and phase error were compared to the observations and presented quantitatively. It was demonstrated that, on average, the source model based on deep ocean measurements can provide better estimation for the wave amplitude (Chapter 5-Section 5.4).

A sensitivity experiment using different spatial resolutions and slope limiters related to the finite volume scheme was carried out. In this regard, resolutions from 450 m to 3600 m were selected and for each resolution, the simulation was repeated for the minmod, Sweby, and Superbee limiters and also using a first-order accurate scheme (Chapter 5-Figure 5.14-5.35). It was shown that for any specific resolution the minmod limiter was the most dissipative compared to other slope limiters, while being the most stable for all simulations. The Sweby limiter could simulate the wave amplitude properly for the coarse resolutions equivalent to the simulations using the minmod limiter with finer resolution. Hence, it was concluded that the Sweby limiter could deliver reliable simulations when using relatively coarse resolutions between 1-2 km, and this saved significant computational time compared with using fine resolution with the minmod limiter. It was further shown that the simulations employing the Superbee limiter with resolutions of 450 m or more became unstable, which means that the Superbee limiter cannot be a reliable option for all cases due to numerical instability.

The simulations considering first-order accuracy appeared to be very dissipative, especially for coarser resolutions. Therefore, the first-order accurate scheme is not recommended for tsunami simulations; at least a second-order accurate numerical scheme must be implemented so that the model is capable to capture the sharply varying tsunami waves properly.

As an important aim of this research, the performance of the GPU hydrodynamic model for a real tsunami simulation was evaluated compared to the sequential model running on a single CPU (Chapter 5-Section 5.6). In this regard, the simulation was carried out by different resolutions across two different types of GPUs as mentioned in the previous subsection 7.1.1. It was concluded that one hour of tsunami wave propagation can be predicted in less than one minute by using the GPU parallel model with resolutions of 1350 m or larger, whereas the same simulation might take about 45 minutes (for the 1350 m resolution) in the sequential code running on a single CPU. This means that the GPU parallel model could reduce the computational time by over 50 times compared to the CPU sequential model. By employing finer resolutions, the performance of the parallel model increased further due to the more effective use of GPU cores. In this regard, simulations involving 12.15 million cells (450 m resolution) take more than 28 hours on CPU, whereas the GPU model only needs about 25 minutes, giving a speed-up of about 70 times. It was also demonstrated that the performance of a regular low-cost GPU running on a PC is comparable to a more expensive GPU running on a server. Most of the simulations could be executed on the desktop GPU; however, for the simulation involving 12.15 million cells the model could only be executed on the server GPU (6 GB) due to the memory limitations of the desktop GPU (1 GB).

The proposed computationally efficient tsunami model is expected to provide a new practical tool for tsunami modelling for different purposes, including real-time warning, evacuation planning, risk management and city planning.

7.1.3 *Adaptive mesh refinement (AMR) techniques*

The adaptive mesh refinement was also implemented for the CPU model and the GPU model individually. The AMR technique used on CPU is based on a simplified dynamic adaptive grid wherein the grids can adapt to the flow features during the simulation (Chapter 6-Section 6.2).

It was then discussed that certain range of GPUs, because of their architecture, have limitations to implement dynamic adaptive algorithms efficiently, while the new generation of GPUs with compute capability of version 3.0 and above can facilitate dynamic parallelization, e.g. GPUs with Kepler microarchitecture (Chapter 6-Section 6.3). In this work, a static adaptive grid was

employed which can be implemented for a wider range of GPUs. It is possible to embed different resolutions, for example, for the near shore region finer grids are considered while in the deep ocean the coarse background resolution is sufficient to provide an appropriate simulation for the lifespan of the tsunami. It was shown that, by using the adaptive technique, fewer cells are needed compared to the uniform grid, which can consequently save the computational time.

The AMR techniques introduced in this work for CPU and GPU were first validated by tsunami benchmarks and further employed to show their capability for the 2011 Japan tsunami. In all cases, it was shown that by considering two levels of refinement, both AMR models could reduce the computational time by about three times compared to the same simulation on high-resolution uniform grids, and the simulated results implementing AMR grids perform to an acceptable degree of accuracy compared to their high-resolution uniform grid counterpart (Chapter 6-Section 6.4).

7.2 Future work and recommendations

Although a substantial effort has been spent to develop, validate and evaluate the GPU hydrodynamic model, there is still work left to fully develop the model to achieve a comprehensive tsunami package. The following subsections propose some potential improvements which should be considered.

7.2.1 *Hydrodynamic model*

In numerical simulations, it is usual that testing and evaluating the model for a wider range of test cases will increase its reliability for future applications. Hence, validating more tsunami benchmarks, especially reproducing other real world case tsunamis e.g. the 2004 Indian Ocean tsunami, is recommended.

The current tsunami model is more suitable for near-field tsunamis as the model has been developed based on a Cartesian grid system. Therefore, providing an updated version of the current model which can run on spherical coordinates is proposed for future work. Then the model is expected to be capable of global scale tsunami simulations.

As the dispersion effect may become important for far-field tsunami simulations, it is recommended to include dispersion in the SWEs without sacrificing the computational efficiency. Some techniques as discussed in Chapter 2 can resemble the physical wave dispersions using higher order numerical terms generated by the finite difference approximations of the shallow water equations. These techniques are also used by some well-

known tsunami packages (e.g. the MOST tsunami model). These approaches are appropriate as they can deliver an acceptable degree of accuracy without affecting the computational efficiency compared with the Boussinesq or 3D models. However, herein the hydrodynamic model is solved by a finite volume scheme rather than a finite difference scheme, and adding the numerical dispersion might be more challenging. Therefore finding a way to couple the numerical dispersion term of the finite difference scheme to the finite volume approach as a hybrid model can potentially improve the capability of the model in an efficient way.

The Coriolis force also may have non-negligible effects on the distribution of the wave height for the far field simulations as discussed in Chapter 2. This term can be simply added to the source term of the SWEs and can be solved explicitly by the proposed finite volume scheme.

Furthermore, there are certain physical features which may be integrated to the depth averaged models. For example, the current hydrodynamic model only considers one layer of water flow (clean water) in the mathematical model. However, in tsunami simulation the flow reaching the land may become two-phase flows composing of granular flow mixtures, applicable to debris-flow floods which may be a new research directly for consideration in the future.

The model may be improved by additional model component to estimate the tsunami forces on coastal structures and buildings. In this regard, new model components must be developed for estimating the hydrostatic and hydrodynamic pressure to quantify the flow impact on structures which can be computed using the flow variables obtained through solving the SWEs. These forces, which may include hydrodynamic force, impulsive force, hydrostatic force, etc., are responsible for the damages to the structures. In order to protect coastal regions and save lives, structures and buildings' design and codes should be improved for the coastlines under risk of tsunamis. The hydrodynamic model can quantify these forces which can be useful for tsunami risk management and practical engineering applications. Therefore for the future work an intensive investigation of a large scale tsunami impact on structures is recommended.

This work has mainly focused on modelling of the tsunami wave propagation caused by the 2011 Tohoku earthquake. The validated hydrodynamic model using different source models can then be employed to estimate the tsunami run-up height and tsunami inundation along the Japanese coast for the next stage of the study.

As discussed in Chapter 2-Section 2.5, since tsunami induced currents can cause major damage to coasts and coastal infrastructure, the simulated velocity estimates should be incorporated into future tsunami predictions. Besides, the high sensitivity of tsunami current velocities to spatial

variability is potentially more challenging for accurate prediction of velocities at specific locations compared to surface elevation estimations, which should be considered for further validation of the proposed tsunami model. The accurate knowledge of tsunami flow velocities and accelerations can also provide necessary information to study sediment transport and tsunami deposits. Simulation of sediment transport is an important future direction in tsunami modeling. Tsunami erosion and deposition modelling could help in improvements of tsunami hazard assessments and it is an essential tool to recreate past events from tsunami deposits (Apotsos *et al.*, 2011).

The hydrodynamic model could be further developed to model the tsunami sediment transport and to estimate the morphological changes associated with erosion and deposition. This would be a multidisciplinary research which will fill the gap between geological investigations and pragmatic implementation of tsunami deposits (Sugawara *et al.*, 2014). Runup and rundown in the coastal region potentially can mobilize the sediment deposits. This may cause damage to building foundations and other coastal infrastructures. Maximum flood extents, wave forces and scour around structures are important to be studied for tsunami engineering designing. Numerical tools beside the experimental methods can help communities to better understand the effects of tsunamis on structures and its effects on sediment deposits. Therefore developing the scour model based on a coupled shallow flow model and sediment transport model is recommended.

7.2.2 *Source model*

As discussed in the literature, there are different approaches for source estimation such as teleseismic waves, GPS data, tsunameter inversions and the combination of these methods. It is recommended to apply different initializations to the current hydrodynamic model and evaluate the accuracy of these source models. Specially, employing the GPS technology results is recommended in order to study the accuracy of this approach for tsunami simulations compared to other source models.

In this work, an impulsive fault rupture is assumed which means the seafloor deforms instantaneously. However, in reality, the sea floor deformation and rupture along the fault may follow a transient process which can be simulated by transient models. This phenomenon might not influence the wave height estimation significantly, because the whole rupture occurs in a short time (few minutes) and the rupture speed of the fault is much larger (an order of magnitude) than the tsunami wave speed. The transient fault model can be also implemented in the current hydrodynamic model to study and evaluate the effect of transient fault on wave propagation compared to the impulsive fault model as used in this work.

It was confirmed that the current hydrodynamic model which takes advantage of the GPU technology can simulate the wave propagation for each simulation in a short time. Therefore, forming stochastic random-field models for an earthquake and conducting an intensive sensitivity analysis of tsunami hazards in relation to the uncertainties of fault and slip geometry will become very convenient with the current GPU model. Similar sensitivity analysis of the source can be combined with the sensitivity experiment which has been attempted in this work using different limiters and special resolutions. These kinds of analysis may require thousands of simulations and the proposed model can be expected to complete the simulation in a sensible time. For example, to carry out such a sensitivity analysis assume that 1,000 separate simulations are needed to be executed. If each simulation can be computed in few seconds or minutes, then the whole analysis might be fulfilled in a day or so. Hence, this task would be interesting to be done in the future to see how the simulated results are affected by the source and the hydrodynamic model simultaneously. These types of simulations are very useful for uncertainty modelling and tsunami hazard and risk mapping. For example, recently Goda and Song (2016) carried out a rigorous computational framework with earthquake source characterization and stochastic simulation of slip distribution. This framework considers various stochastic tsunami scenarios to estimate the uncertainty propagation related to the source characteristics in a probabilistic risk analysis. In this regard, they employed an intensive Monte Carlo tsunami simulation for the 2011 Tohoku tsunami using 726 stochastic slip models derived from 11 source models. Herein, the sensitivity analysis by combination of the source model characteristics (e.g., fault geometry, slip distribution) and numerical features (e.g., grid size, slope limiter, friction) is recommended. These kind of tedious simulations can be carried out much more conveniently by the current model.

7.2.3 *Tsunami measurement technologies*

Recently, technological development has been promoted to predict tsunami features at the earliest time. Tsunami measurements are usually carried out by tide gauge or bottom pressure sensor, or kinematic GPS buoys. The most conventional method is to measure by tide gauge, whereas the modern procedure is to operate GPS buoys and bottom pressure technologies. Traditional tide gauges at the coast are not suitable for early tsunami detections because of their deployed location (Matsumoto, 2011). Deploying DART station in deep-ocean known to be effective for measuring the tsunami heights (González *et al.*, 2005). However, this technology may face operational challenges which requires huge expenses. Manley and Hine (2011) deployed new wave glider technology at DART station (#46412) off the coast of San Diego to assess the accuracy of this new approach. The data relayed from the wave glider reported to be

in exact agreement with conventional mooring data at DART station. Wave glider is easy to be deployed with low cost of operations and long-range endurance which will enable new approaches to undersea operations, the research for different applications is underway. For example, Carragher *et al.* (2013) used wave glider sensor for variety scientific missions and applications such as oilfield exploration and tsunami measurements.

More recently Morales Maqueda *et al.* (2016) successfully applied kinematic precise point positioning to GPS data from a Wave Glider to determine the centimeter-precise of water height for the Loch Ness lake in Scotland. They mentioned that the wave gliders have the potential to determine water surface heights with centimeter precision globally for lake and ocean modelling. Also it is stated this novel GPS wave glider technology can be deployed autonomously for the regions where not accessible for conventional stations such as GPS buoys, tide gauges and bottom pressure recorders. According to the advantages of this inexpensive and promising technology which is less reported for tsunami applications, it would be interesting to more investigate the application of the recent technology for accurate and efficient tsunami predictions.

7.2.4 *Parallelization*

This work introduced a computationally efficient tsunami model which employs GPU technology. By evaluating the model performance it was evident that the algorithm implemented for the hydrodynamic model was suitable for parallelization. However, in this work the parallel model was developed to run on a single GPU. On the other hand, for very large scale simulations (e.g., global scale), especially if higher-resolution grids are also required to be considered in one simulation, then the single GPU might not be capable of handling the simulations due to memory limitations. Therefore, for the future work it is recommended to employ a hybrid parallelization (MPI+CUDA) using the model decomposition technique as discussed in Chapter 2. Such a parallel model would be suitable to be executed on GPU clusters where each discretized subdomain will be executed by one GPU and the information in the interface of the boundaries between subdomains is passed and communicated by the MPI system. This fully parallelized model can overcome the memory shortage, and the model performance according to the number of GPUs potentially might scale up linearly.

As a more challenging part would be implementing the adaptive refinement technique (AMR) with the fully parallelized model. Model decomposition on uniform grids is relatively easier as the load balance can be distributed uniformly between processors, however, the computational

load for AMR grids should be redistributed between processors, which should be addressed in the future work.

Currently, tsunami forecast models by NOAA are installed on multiprocessor supercomputers of the type which are used for weather forecasting, to provide an inundation forecast for coastal communities within 10 minutes of source determination (Tang *et al.*, 2009). Heterogeneous parallel computing using GPUs and CPUs as introduced in this work is another option under consideration. It is expected the computational time of the forecast models can be reduced to few minutes or even seconds by powerful computers.

References

Abadie, S.M., Harris, J.C., Grilli, S.T. and Fabre, R. (2012) 'Numerical modeling of tsunami waves generated by the flank collapse of the Cumbre Vieja Volcano (La Palma, Canary Islands): tsunami source and near field effects', *Journal of Geophysical Research: Oceans*, 117(C5).

Abe, K. (1979) 'Size of great earthquakes of 1837–1974 inferred from tsunami data', *Journal of Geophysical Research: Solid Earth*, 84(B4), pp. 1561-1568.

Abe, K. (2011) 'Synthesis of a tsunami spectrum in a semi-enclosed basin using its background spectrum', *Pure and applied geophysics*, 168(6-7), pp. 1101-1112.

Acuña, M. and Aoki, T. (2009) 'Real-time tsunami simulation on multi-node GPU cluster', *ACM/IEEE conference on supercomputing*.

Agbaglah, G., Delaux, S., Fuster, D., Hoepffner, J., Josserand, C., Popinet, S., Ray, P., Scardovelli, R. and Zaleski, S. (2011) 'Parallel simulation of multiphase flows using octree adaptivity and the volume-of-fluid method', *Comptes Rendus Mecanique*, 339(2), pp. 194-207.

Amdahl, G.M. (1967) 'Validity of the single processor approach to achieving large scale computing capabilities', *Proceedings of the April 18-20, 1967, spring joint computer conference*. ACM, pp. 483-485.

Ammon, C.J., Lay, T., Kanamori, H. and Cleveland, M. (2011) 'A rupture model of the 2011 off the Pacific coast of Tohoku Earthquake', *Earth, Planets and Space*, 63(7), pp. 693-696.

Amouzgar, R. and Liang, Q. (2014) 'Application of a GPU Based Hydrodynamic Model in Tsunami Simulations', Busan, Korea, 15-20 June. ISOPE: 24th International Society of Offshore and Polar Engineers.

Amouzgar, R., Liang, Q., Clarke, P.J., Yasuda, T. and Mase, H. (2016) 'Computationally Efficient Tsunami Modeling on Graphics Processing Units (GPUs)', *International Journal of Offshore and Polar Engineering*, 26(02), pp. 154-160.

Amouzgar, R., Liang, Q. and Smith, L. (2014) 'A GPU-accelerated shallow flow model for tsunami simulations', *Proceedings of the ICE - Engineering and Computational Mechanics*, 167(3), pp. 117-125.

Apotsos, A., Buckley, M., Gelfenbaum, G., Jaffe, B. and Vatvani, D. (2011) 'Nearshore Tsunami Inundation Model Validation: Toward Sediment Transport Applications', *Pure and Applied Geophysics*, 168(11), pp. 2097-2119.

Arcos, M.E.M. and LeVeque, R.J. (2015) 'Validating velocities in the GeoClaw tsunami model using observations near Hawaii from the 2011 Tohoku tsunami', *Pure and Applied Geophysics*, 172(3-4), pp. 849-867.

Audusse, E., Bouchut, F., Bristeau, M.O., Klein, R. and Perthame, B. (2004) 'A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows', *SIAM Journal on Scientific Computing*, 25(6), pp. 2050-2065.

Behrens, J., Androsov, A., Braune, S., Danilov, S., Harig, S., Schröter, J., Sein, D.V., Sidorenko, D., Startseva, O. and Taguchi, E. (2007) 'TsunAWI technical documentation part I: mathematical, numerical and implementation concepts', *Tsunami Project Documentation*, (004).

Berger, M.J. and Colella, P. (1989) 'Local adaptive mesh refinement for shock hydrodynamics', *Journal of computational Physics*, 82(1), pp. 64-84.

Berger, M.J., George, D.L., LeVeque, R.J. and Mandli, K.T. (2011) 'The GeoClaw software for depth-averaged flows with adaptive refinement', *Advances in Water Resources*, 34(9), pp. 1195-1206.

Berger, M.J. and Oliger, J. (1984) 'Adaptive mesh refinement for hyperbolic partial differential equations', *Journal of Computational Physics*, 53(3), pp. 484-512.

Bernard, E. and Titov, V. (2015) 'Evolution of tsunami warning systems and products', *Phil. Trans. R. Soc. A*, 373(2053), p. 20140371.

Bernard, E., Wei, Y., Tang, L. and Titov, V. (2014) 'Impact of near-field, deep-ocean tsunami observations on forecasting the 7 December 2012 Japanese tsunami', *Pure and Applied Geophysics*, 171(12), pp. 3483-3491.

Bernard, E.N., Meinig, C., Titov, V.V., O'Neil, K., Lawson, R., Jarrott, K., Bailey, R., Nelson, F., Tinti, S., von Hillebrandt, C. and Koltermann, P. (2010) 'Tsunami resilient communities', *Proceedings of the OceanObs09: Sustained Ocean Observations and Information for Society Conference*.

Blewitt, G., Kreemer, C., Hammond, W.C., Plag, H.P., Stein, S. and Okal, E. (2006) 'Rapid determination of earthquake magnitude using GPS for tsunami warning systems', *Geophysical Research Letters*, 33(11).

Breuer, A. and Bader, M. (2012) 'Teaching Parallel Programming Models on a Shallow-Water Code', *11th International Symposium on Parallel and Distributed Computing*. 25-29 June 2012. pp. 301-308.

Briggs, M., Synolakis, C., Harkins, G. and Green, D. (1995) 'Laboratory experiments of tsunami runup on a circular island', *pure and applied geophysics*, 144(3-4), pp. 569-593.

Brodtkorb, A.R. (2010) *Scientific computing on heterogeneous architectures*. PhD thesis: University of Oslo.

Brodtkorb, A.R., Hagen, T.R., Lie, K.A. and Natvig, J.R. (2010) 'Simulation and visualization of the Saint-Venant system using GPUs', *Computing and Visualization in Science*, 13(7), pp. 341-353.

Brodtkorb, A.R., Sætra, M.L. and Altinakar, M. (2012) 'Efficient shallow water simulations on GPUs: Implementation, visualization, verification, and validation', *Computers and Fluids*, 55, pp. 1-12.

Brufau, P., García-Navarro, P. and Vázquez-Cendón, M.E. (2004) 'Zero mass error using unsteady wetting-drying conditions in shallow flows over dry irregular topography', *International Journal for Numerical Methods in Fluids*, 45(10), pp. 1047-1082.

Brufau, P., Vzquez-Cendon, M.E. and Garca-Navarro, P. (2002) 'A numerical model for the flooding and drying of irregular domains', *International Journal for Numerical Methods in Fluids*, 39(3), pp. 247-275.

Bryan, G.L., Norman, M.L., O'Shea, B.W., Abel, T., Wise, J.H., Turk, M.J., Reynolds, D.R., Collins, D.C., Wang, P. and Skillman, S.W. (2014) 'Enzo: An adaptive mesh refinement code for astrophysics', *The Astrophysical Journal Supplement Series*, 211(2), p. 19.

Burwell, D., Tolkova, E. and Chawla, A. (2007) 'Diffusion and dispersion characterization of a numerical tsunami model', *Ocean Modelling*, 19(1), pp. 10-30.

Bussing, T.R.A. and Murman, E.M. (1988) 'Finite-volume method for the calculation of compressible chemically reacting flows', *Aiaa Journal*, 26(9), pp. 1070-1078.

Campbell, P.M., Devine, K.D., Flaherty, J.E., Gervasio, L.G. and Teresco, J.D. (2003) 'Dynamic octree load balancing using space-filling curves', *Williams College Department of Computer Science, Technical Report CS-03*, 1, p. 68.

Carragher, P., Hine, G., Legh-Smith, P., Mayville, J., Nelson, R., Pai, S., Parnum, I., Shone, P., Smith, J. and Tichatschke, C. (2013) 'A New Platform for Offshore Exploration and Production', *Oilfield Review*, 2014(25), p. 4.

Castro, M.J., Ortega, S., De La Asunción, M. and Mantas, J.M. (2010) 'On the benefits of using GPUs to simulate shallow flows with finite volume schemes', *SeMA Journal*, 50(1), pp. 27-44.

Chan, I. and Liu, P.L.F. (2012) 'On the runup of long waves on a plane beach', *Journal of Geophysical Research: Oceans*, 117(C8).

Chandra, R. (2001) *Parallel programming in OpenMP*. San Francisco: Morgan Kaupmann Publishers.

Chen, Q., Kirby, J.T., Dalrymple, R.A., Kennedy, A.B. and Chawla, A. (2000) 'Boussinesq modeling of wave transformation, breaking, and runup. II: 2D', *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 126(1), pp. 48-56.

Cheung, K.F., Bai, Y. and Yamazaki, Y. (2013) 'Surges around the Hawaiian Islands from the 2011 Tohoku tsunami', *Journal of Geophysical Research: Oceans*, 118(10), pp. 5703-5719.

Choi, B.H., Kim, D.C., Pelinovsky, E. and Woo, S.B. (2007) 'Three‐dimensional simulation of tsunami run‐up around conical island', *Coastal Engineering*, 54(8), pp. 618-629.

Clarke, P.J., Paradissis, D., Briole, P., England, P.C., Parsons, B.E., Billiris, H., Veis, G. and Ruegg, J.C. (1997) 'Geodetic investigation of the 13 May 1995 Kozani-Grevena (Greece) earthquake', *Geophysical Research Letters*, 24(6), pp. 707-710.

Courant, R., Friedrichs, F. and Lewy, H. (1967) 'On the Partial Difference Equations of Mathematical Physics', *IBM Journal of Research and Development*, 11(2), pp. 215-234.

Cowles, G.W. (2013) 'A Block-Structured Adaptive Mesh Refinement Solver for Morphodynamic Modeling', *Journal of Coastal Research*, pp. 727-735.

Crespo, A.C., Dominguez, J.M., Barreiro, A., Gómez-Gesteira, M. and Rogers, B.D. (2011) 'GPUs, a new tool of acceleration in CFD: Efficiency and reliability on smoothed particle hydrodynamics methods', *PLoS ONE*, 6(6).

CUDA Toolkit (2013) *CUDA Toolkit Documentation*. Available at: http://docs.nvidia.com/cuda/thrust/index.html (Accessed: 26/07/ 2016).

Danalis, A., Marin, G., McCurdy, C., Meredith, J.S., Roth, P.C., Spafford, K., Tipparaju, V. and Vetter, J.S. (2010) 'The Scalable Heterogeneous Computing (SHOC) benchmark suite', *Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units*. Pittsburgh, Pennsylvania, USA. ACM.

de la Asunción, M., Castro, M.J., Mantas, J.M. and Ortega, S. (2016) 'Numerical simulation of tsunamis generated by landslides on multiple GPUs', *Advances in Engineering Software*, 99, pp. 59-72.

Delis, A.I. and Mathioudakis, E.N. (2009) 'A finite volume method parallelization for the simulation of free surface shallow water flows', *Mathematics and Computers in Simulation*, 79(11), pp. 3339-3359.

Desai, B., Maskrey, A., Peduzzi, P., De Bono, A. and Herold, C. (2015) 'Making Development Sustainable: The Future of Disaster Risk Management, Global Assessment Report on Disaster Risk Reduction'.

Domínguez, J.M., Crespo, A.J.C., Valdez-Balderas, D., Rogers, B.D. and Gómez-Gesteira, M. (2013) 'New multi-GPU implementation for smoothed particle hydrodynamics on heterogeneous clusters', *Computer Physics Communications*, 184(8), pp. 1848-1860.

Erduran, K.S., Kutija, V. and Hewett, C.J.M. (2002) 'Performance of finite volume solutions to the shallow water equations with shock-capturing schemes', *International Journal for Numerical Methods in Fluids*, 40(10), pp. 1237-1273.

Eze, C.L., Uko, D.E., Gobo, A.E., Sigalo, F.B. and Israel-Cookey, C. (2009) 'Mathematical Modelling of Tsunami Propagation', *Journal of Applied Sciences and Environmental Management*, 13(3).

Fiedler, F.R. and Ramirez, J.A. (2000) 'A numerical method for simulating discontinuous shallow flow over an infiltrating surface', *International Journal for Numerical Methods in Fluids*, 32(2), pp. 219-240.

Fraccarollo, L. and Toro, E.F. (1995) 'Experimental and numerical assessment of the shallow water model for two-dimensional dam-break type problems', *Journal of Hydraulic Research/De Recherches Hydrauliques*, 33(6), pp. 843-864.

Franchello, G. (2010) 'Shoreline tracking and implicit source terms for a well balanced inundation model', *International Journal for Numerical Methods in Fluids*, 63(10), pp. 1123-1146.

Fringer, O.B., Gerritsen, M. and Street, R.L. (2006) 'An unstructured-grid, finite-volume, nonhydrostatic, parallel coastal ocean simulator', *Ocean Modelling*, 14(3), pp. 139-173.

Fritz, H.M., Phillips, D.A., Okayasu, A., Shimozono, T., Liu, H., Mohammed, F., Skanavis, V., Synolakis, C.E. and Takahashi, T. (2012) 'The 2011 Japan tsunami current velocity measurements from survivor videos at Kesennuma Bay using LiDAR', *Geophysical Research Letters*, 39(7).

Fujii, Y., Satake, K., Sakai, S.i., Shinohara, M. and Kanazawa, T. (2011) 'Tsunami source of the 2011 off the Pacific coast of Tohoku Earthquake', *Earth Planets and Space*, 63(7), pp. 815-820.

Fujiwara, T., Kodaira, S., No, T., Kaiho, Y., Takahashi, N. and Kaneda, Y. (2011) 'The 2011 Tohoku-Oki earthquake: Displacement reaching the trench axis', *Science*, 334(6060), pp. 1240-1240.

Funke, S.W., Pain, C.C., Kramer, S.C. and Piggott, M.D. (2011) 'A wetting and drying algorithm with a combined pressure/free-surface formulation for non-hydrostatic models', *Advances in Water Resources*, 34(11), pp. 1483-1495.

Gandham, R., Medina, D. and Warburton, T. (2015) 'GPU Accelerated Discontinuous Galerkin Methods for Shallow Water Equations', *Communications in Computational Physics*, 18(1), pp. 37-64.

George, D.L. (2006) *Finite volume methods and adaptive refinement for tsunami propagation and inundation*. PhD thesis. University of Washington.

George, D.L. and LeVeque, R.J. (2006) 'Finite volume methods and adaptive refinement for global tsunami propagation and local inundation', *Science of Tsunami Hazards*, 24, pp. 319-328.

George, D.L. and LeVeque, R.J. (2008) 'High-resolution methods and adaptive refinement for tsunami propagation and inundation', in *Hyperbolic Problems: Theory, Numerics, Applications*. Springer, pp. 541-549.

Gica, E., Spillane, M.C., Titov, V.V., Chamberlin, C.D. and Newman, J.C. (2008) *Development of the Forecast Propagation Database for NOAA's Short-Term Inundation Forecast for Tsunamis(SIFT), NOAA Tech. Memo. OAR PMEL-139, 89 pp*. Seattle.

Gidra, H., Haque, I., Kumar, N.P., S, M., Gaur, M.S., Laxmi, V., Zwolinski, M. and Singh, V. (2011) *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on*. 2-4 Sept. 2011.

Gingold, R.A. and Monaghan, J.J. (1977) 'Smoothed particle hydrodynamics: theory and application to non-spherical stars', *Monthly notices of the royal astronomical society*, 181(3), pp. 375-389.

Glimsdal, S., Pedersen, G.K., Harbitz, C.B. and Løvholt, F. (2013) 'Dispersion of tsunamis: does it really matter?', *Nat. Hazards Earth Syst. Sci*, 13, pp. 1507-1526.

Goda, K., Mai, P.M., Yasuda, T. and Mori, N. (2014) 'Sensitivity of tsunami wave profiles and inundation simulations to earthquake slip and fault geometry for the 2011 Tohoku earthquake', *Earth, Planets and Space*, 66(1), pp. 1-20.

Goda, K. and Song, J. (2016) 'Uncertainty modeling and visualization for tsunami hazard and risk mapping: a case study for the 2011 Tohoku earthquake', *Stochastic Environmental Research and Risk Assessment*, pp. 1-15.

González, F.I., Bernard, E.N., Meinig, C., Eble, M.C., Mofjeld, H.O. and Stalin, S. (2005) 'The NTHMP tsunameter network', *Natural Hazards*, 35(1), pp. 25-39.

González, F.I., LeVeque, R.J., Chamberlain, P., Hirai, B., Varkovitzky, J. and George, D.L. (2011) 'Validation of the GeoClaw Model', *Grupo de Modelación Geoclaw Tsunami, Universidad de Washington*.

Goto, C. and Ogawa, Y. (1982) 'Tsunami numerical simulation with leapfrog scheme', *Tohoku Univ*, p. 52.

Goto, C., Ogawa, Y., Shuto, N. and Imamura, F. (1997) *IUGG/IOC time project: Numerical method of tsunami simulation with the leap-frog scheme*. Unesco Paris, Frace.

Grilli, S.T., Harris, J.C., Tajalli Bakhsh, T.S., Masterlark, T.L., Kyriakopoulos, C., Kirby, J.T. and Shi, F. (2012) 'Numerical Simulation of the 2011 Tohoku Tsunami Based on a New Transient FEM Co-seismic Source: Comparison to Far- and Near-Field Observations', *Pure and Applied Geophysics*, 170(6), pp. 1333-1359.

Grilli, S.T., Ioualalen, M., Asavanant, J., Shi, F., Kirby, J.T. and Watts, P. (2007) 'Source constraints and model simulation of the December 26, 2004, Indian Ocean tsunami', *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 133(6), pp. 414-428.

Grudenić, I., Groš, S. and Bogunović, N. (2008) 'Load balancing MPI algorithm for high throughput applications', *Information Technology Interfaces, 2008. ITI 2008. 30th International Conference on*. IEEE, pp. 839-844.

Gusman, A.R., Tanioka, Y., Sakai, S. and Tsushima, H. (2012) 'Source model of the great 2011 Tohoku earthquake estimated from tsunami waveforms and crustal deformation data', *Earth and Planetary Science Letters*, 341, pp. 234-242.

Harig, S., Chaeroni, Pranowo, W.S. and Behrens, J. (2008) 'Tsunami simulations on several scales', *Ocean Dynamics*, 58(5), pp. 429-440.

Harten, L., Lax, P. and van Leer, B. (1983) 'On upstream differencing and Godunov-type schemes for hyperbolic conservation laws', *SIAM Review*, 25(1), pp. 35-61.

Hayes, G.P. (2011) 'Rapid source characterization of the 2011 M w 9.0 off the Pacific coast of Tohoku Earthquake', *Earth, planets and space*, 63(7), pp. 529-534.

Hirsch, C. (1995) *Numercal computation of internal and external flows Volume 2: computational methods for inviscid and viscous*. New York, USA: Wiley.

Hooper, A., Pietrzak, J., Simons, W., Cui, H., Riva, R., Naeije, M., van Scheltinga, A.T., Schrama, E., Stelling, G. and Socquet, A. (2013) 'Importance of horizontal seafloor motion on tsunami height for the 2011 M-w=9.0 Tohoku-Oki earthquake', *Earth and Planetary Science Letters*, 361, pp. 469-479.

Horrillo, J., Grilli, S.T., Nicolsky, D., Roeber, V. and Zhang, J. (2015) 'Performance benchmarking tsunami models for NTHMP's inundation mapping activities', *Pure and Applied Geophysics*, 172(3-4), pp. 869-884.

Horrillo, J., Kowalik, Z. and Shigihara, Y. (2006) 'Wave Dispersion Study in the Indian Ocean-Tsunami of December 26, 2004', *Marine Geodesy*, 29(3), pp. 149-166.

Horrillo, J., Wood, A., Kim, G.B. and Parambath, A. (2013) 'A simplified 3 - D Navier - Stokes numerical model for landslide - tsunami: Application to the Gulf of Mexico', *Journal of Geophysical Research: Oceans*, 118(12), pp. 6934-6950.

Hou, J., Liang, Q. and Xia, X. (2015) 'Robust absorbing boundary conditions for shallow water flow models', *Environmental Earth Sciences*, 74(11), pp. 7407-7422.

Hou, J., Simons, F., Mahgoub, M. and Hinkelmann, R. (2013) 'A robust well-balanced model on unstructured grids for shallow water flows with wetting and drying over complex topography', *Computer Methods in Applied Mechanics and Engineering*, 257, pp. 126-149.

Hu, K., Mingham, C.G. and Causon, D.M. (2000) 'Numerical simulation of wave overtopping of coastal structures using the non-linear shallow water equations', *Coastal Engineering*, 41(4), pp. 433-465.

Hubbard, M.E. and Dodd, N. (2002) 'A 2D numerical model of wave run-up and overtopping', *Coastal Engineering*, 47(1), pp. 1-26.

Huppert, H.E. and Sparks, R.S.J. (2006) 'Extreme natural hazards: population growth, globalization and environmental change', *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 364(1845), pp. 1875-1888.

Ide, S., Baltay, A. and Beroza, G.C. (2011) 'Shallow dynamic overshoot and energetic deep rupture in the 2011 Mw 9.0 Tohoku-Oki earthquake', *Science*, 332(6036), pp. 1426-1429.

Iinuma, T., Hino, R., Kido, M., Inazu, D., Osada, Y., Ito, Y., Ohzono, M., Tsushima, H., Suzuki, S. and Fujimoto, H. (2012) 'Coseismic slip distribution of the 2011 off the Pacific Coast of Tohoku Earthquake (M9. 0) refined by means of seafloor geodetic data', *Journal of Geophysical Research: Solid Earth*, 117(B7).

Imamura, F. (1996) 'Review of tsunami simulation with a finite difference method', *Long-wave runup models*, pp. 25-42.

Imamura, F., Shuto, N. and Goto, C. (1988) *Sixth Congress of the Asian and Pacific Regional Division, Int. Assoc. Hydraul. Res., Kyoto, Japan*.

Ioualalen, M., Asavanant, J., Kaewbanjak, N., Grilli, S.T., Kirby, J.T. and Watts, P. (2007) 'Modeling the 26 December 2004 Indian Ocean tsunami: Case study of impact in Thailand', *Journal of Geophysical Research: Oceans*, 112(C7).

Ito, Y., Tsuji, T., Osada, Y., Kido, M., Inazu, D., Hayashi, Y., Tsushima, H., Hino, R. and Fujimoto, H. (2011) 'Frontal wedge deformation near the source region of the 2011 Tohoku-Oki earthquake', *Geophysical Research Letters*, 38(7).

Ji, C., Wald, D.J. and Helmberger, D.V. (2002) 'Source description of the 1999 Hector Mine, California, earthquake, part I: Wavelet domain inversion theory and resolution analysis', *Bulletin of the Seismological Society of America*, 92(4), pp. 1192-1207.

Ji, H., Lien, F.-S. and Yee, E. (2010) 'A new adaptive mesh refinement data structure with an application to detonation', *Journal of Computational Physics*, 229(23), pp. 8981-8993.

Juez, C., Lacasta, A., Murillo, J. and García-Navarro, P. (2016) 'An efficient GPU implementation for a faster simulation of unsteady bed-load transport', *Journal of Hydraulic Research*, pp. 1-14.

Kao, H.-M. and Chang, T.-J. (2012) 'Numerical modeling of dambreak-induced flood and inundation using smoothed particle hydrodynamics', *Journal of hydrology*, 448, pp. 232-244.

Kawahara, M. and Umetsu, T. (1986) 'Finite element method for moving boundary problems in river flow', *International Journal for Numerical Methods in Fluids*, 6(6), pp. 365-386.

Kazolea, M., Delis, A.I., Nikolos, I.K. and Synolakis, C.E. (2012) 'An unstructured finite volume numerical scheme for extended 2D Boussinesq-type equations', *Coastal Engineering*, 69, pp. 42-66.

Kesserwani, G. and Liang, Q.H. (2010) 'A discontinuous Galerkin algorithm for the two-dimensional shallow water equations', *Computer Methods in Applied Mechanics and Engineering*, 199(49-52), pp. 3356-3368.

Kim, D.-H. and Lynett, P.J. (2010) 'Dispersive and nonhydrostatic pressure effects at the front of surge', *Journal of Hydraulic Engineering*, 137(7), pp. 754-765.

Kim, D.-H., Lynett, P.J. and Socolofsky, S.A. (2009) 'A depth-integrated model for weakly dispersive, turbulent, and rotational fluid flows', *Ocean Modelling*, 27(3), pp. 198-214.

Kim, D.C., Kim, K.O., Pelinovsky, E., Didenkulova, I. and Choi, B.H. (2013) 'Three-dimensional tsunami runup simulation for the port of Koborinai on the Sanriku coast of Japan', *Journal of coastal research*, 1(65), p. 266.

Kirby, J.T., Pophet, N., Shi, F. and Grilli, S.T. (2009) 'Basin scale tsunami propagation modeling using Boussinesq models: Parallel implementation in spherical coordinates', *Proc. WCCE-ECCE-TCCE Joint Conf. on Earthquake and Tsunami (Istanbul, Turkey, June 22-24)*.

Kirby, J.T., Shi, F., Tehranirad, B., Harris, J.C. and Grilli, S.T. (2013) 'Dispersive tsunami waves in the ocean: Model equations and sensitivity to dispersion and Coriolis effects', *Ocean Modelling*, 62, pp. 39-55.

Kirby, J.T., Wei, G., Chen, Q., Kennedy, A.B. and Dalrymple, R.A. (1998) *FUNWAVE 1.0: fully nonlinear Boussinesq wave model-Documentation and user's manual*. University of Delaware.

Kirk, D.B. and Wen-mei, W.H. (2012) *Programming massively parallel processors: a hands-on approach*. Newnes.

Kowalik, Z., Knight, W., Logan, T. and Whitmore, P. (2005) 'Numerical modeling of the global tsunami: Indonesian tsunami of 26 December 2004', *Science of Tsunami Hazards*, 23(1), pp. 40-56.

Kuo, F.A., Smith, M.R., Hsieh, C.W., Chou, C.Y. and Wu, J.S. (2011) 'GPU acceleration for general conservation equations and its application to several engineering problems', *Computers and Fluids*, 45(1), pp. 147-154.

Lacasta, A., Morales-Hernández, M., Murillo, J. and García-Navarro, P. (2014) 'An optimized GPU implementation of a 2D free surface simulation model on unstructured meshes', *Advances in Engineering Software*, 78, pp. 1-15.

Lannes, D. and Marche, F. (2015) 'A new class of fully nonlinear and weakly dispersive Green–Naghdi models for efficient 2D simulations', *Journal of Computational Physics*, 282, pp. 238-268.

Lavrentiev-jr, M., Romanenko, A., Titov, V. and Vazhenin, A. (2009) *Parallel computing technologies*. Novosibirsk, Russia. Springer-Verlag.

Lay, T., Ammon, C.J., Kanamori, H., Xue, L. and Kim, M.J. (2011a) 'Possible large near-trench slip during the 2011 Mw 9.0 off the Pacific coast of Tohoku Earthquake', *Earth, Planets and Space*, 63(7), pp. 687-692.

Lay, T., Yamazaki, Y., Ammon, C.J., Cheung, K.F. and Kanamori, H. (2011b) 'The 2011 M w 9.0 off the Pacific coast of Tohoku Earthquake: Comparison of deep-water tsunami signals with finite-fault rupture model predictions', *Earth, planets and space*, 63(7), pp. 797-801.

Lee, W.-K., Borthwick, A.G.L. and Taylor, P.H. (2011) 'A fast adaptive quadtree scheme for a two-layer shallow water model', *Journal of Computational Physics*, 230(12), pp. 4848-4870.

LeVeque, R.J. (1992) *Numerical Methods for Conservation Laws*. Birkhäuser Basel.

LeVeque, R.J. and George, D.L. (2008) 'High-resolution finite volume methods for the shallow water equations with bathymetry and dry states', *Advanced numerical models for simulating tsunami waves and runup*, 10, pp. 43-73.

Leveque, R.J., George, D.L. and Berger, M.J. (2011) 'Tsunami modelling with adaptively refined finite volume methods', *Acta Numerica*, 20, pp. 211-289.

Liang, Q. (2010) 'Flood Simulation Using a Well-Balanced Shallow Flow Model', *Journal of Hydraulic Engineering - ASCE*, 136(9), pp. 669-675.

Liang, Q. (2012) 'A simplified adaptive Cartesian grid system for solving the 2D shallow water equations', *International Journal for Numerical Methods in Fluids*, 69(2), pp. 442-458.

Liang, Q. and Borthwick, A.G.L. (2009) 'Adaptive quadtree simulation of shallow flows with wet-dry fronts over complex topography', *Computers and Fluids*, 38(2), pp. 221-234.

Liang, Q., Borthwick, A.G.L. and Stelling, G. (2004) 'Simulation of dam- and dyke-break hydrodynamics on dynamically adaptive quadtree grids', *International Journal for Numerical Methods in Fluids*, 46(2), pp. 127-162.

Liang, Q., Hou, J. and Amouzgar, R. (2015a) 'Simulation of Tsunami Propagation Using Adaptive Cartesian Grids', *Coastal Engineering Journal*, p. 1550016.

Liang, Q., Hou, J. and Xia, X. (2015b) 'Contradiction between the C-property and mass conservation in adaptive grid based shallow flow models: cause and solution', *International Journal for Numerical Methods in Fluids*, 78(1), pp. 17-36.

Liang, Q. and Marche, F. (2009) 'Numerical resolution of well-balanced shallow water equations with complex source terms', *Advances in Water Resources*, 32(6), pp. 873-884.

Liang, W.-Y., Hsieh, T.-J., Satria, M.T., Chang, Y.-L., Fang, J.-P., Chen, C.-C. and Han, C.-C. (2009) 'A GPU-Based Simulation of Tsunami Propagation and Inundation', in Hua, A. and Chang, S.-L. (eds.) *Algorithms and Architectures for Parallel Processing: 9th International Conference, ICA3PP 2009, Taipei, Taiwan, June 8-11, 2009. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 593-603.

Liu, P.L.F., Cho, Y.-S., Briggs, M.J., Kanoglu, U. and Synolakis, C.E. (1995a) 'Runup of solitary waves on a circular island', *Journal of Fluid Mechanics*, 302, pp. 259-285.

Liu, P.L.F., Cho, Y.S., Yoon, S.B. and Seo, S.N. (1995b) 'Numerical Simulations of the 1960 Chilean Tsunami Propagation and Inundation at Hilo, Hawaii', in Tsuchiya, Y. and Shuto, N. (eds.) *Tsunami: Progress in Prediction, Disaster Prevention and Warning*. Dordrecht: Springer Netherlands, pp. 99-115.

Liu, P.L.F., Woo, S.B. and Cho, Y.S. (1998) 'Computer programs for tsunami propagation and inundation. School of Civil and Environmental Engineering, Cornell University, USA'.

Liu, P.L.F., Yeh, H. and Synolakis, C. (2008) *Advanced numerical models for simulating tsunami waves and runup*. Singapore: World Scientific Publishing Company.

Liu, X., Xu, H., Shao, S. and Lin, P. (2013) 'An improved incompressible SPH model for simulation of wave–structure interaction', *Computers & Fluids*, 71, pp. 113-123.

Liu, Y., Shi, Y., Yuen, D.A., Sevre, E.O.D., Yuan, X. and Xing, H.L. (2009) 'Comparison of linear and nonlinear shallow wave water equations applied to tsunami waves over the China Sea', *Acta Geotechnica*, 4(2), pp. 129-137.

Løvholt, F., Kaiser, G., Glimsdal, S., Scheele, L., Harbitz, C.B. and Pedersen, G. (2012) 'Modeling propagation and inundation of the 11 March 2011 Tohoku tsunami', *Natural Hazards and Earth System Sciences*, 12(4), pp. 1017-1028.

Løvholt, F., Pedersen, G. and Gisler, G. (2008) 'Oceanic propagation of a potential tsunami from the La Palma Island', *Journal of Geophysical Research: Oceans*, 113(C9).

Løvholt, F., Pedersen, G. and Glimsdal, S. (2010) 'Coupling of Dispersive Tsunami Propagation and Shallow Water Coastal', *Open Oceanography Journal*, 4, pp. 71-82.

Lynett, P. and Liu, P.L.F. (2002) 'A numerical study of submarine–landslide–generated waves and run–up', *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 458, pp. 2885-2910.

Lynett, P. and Liu, P.L.F. (2004) 'A two-layer approach to wave modelling', *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 460(2049), pp. 2637-2669.

Lynett, P., Liu, P.L.F., Sitanggang, K.I. and Kim, D. (2002a) *Modeling wave generation, evolution, and interaction with depth-integrated, dispersive wave equations COULWAVE code manual, Long and Intermediate Wave Modeling Package Cornell University*. Cornell University, Ithaca, USA.

Lynett, P.J., Borrero, J., Son, S., Wilson, R. and Miller, K. (2014) 'Assessment of the tsunami ‐ induced current hazard', *Geophysical Research Letters*, 41(6), pp. 2048-2055.

Lynett, P.J., Borrero, J.C., Liu, P.L.F. and Synolakis, C.E. (2003) 'Field Survey and Numerical Simulations: A Review of the 1998 Papua New Guinea Tsunami', in Bardet, J.-P., Imamura, F., Synolakis, C.E., Okal, E.A. and Davies, H.L. (eds.) *Landslide Tsunamis: Recent Findings and Research Directions*. Basel: Birkhäuser Basel, pp. 2119-2146.

Lynett, P.J., Wu, T.R. and Liu, P.L.F. (2002b) 'Modeling wave runup with depth-integrated equations', *Coastal Engineering*, 46(2), pp. 89-107.

Ma, Z.H., Wang, H. and Pu, S.H. (2014) 'GPU computing of compressible flow problems by a meshless method with space-filling curves', *Journal of Computational Physics*, 263, pp. 113-135.

MacInnes, B.T., Gusman, A.R., LeVeque, R.J. and Tanioka, Y. (2013) 'Comparison of Earthquake Source Models for the 2011 Tohoku Event Using Tsunami Simulations and Near ‐ Field Observations', *Bulletin of the Seismological Society of America*, 103(2B), pp. 1256-1274.

Madsen, O.S. and Mei, C.C. (1969) 'The transformation of a solitary wave over an uneven bottom', *Journal of Fluid Mechanics*, 39(04), pp. 781-791.

Maeda, T., Furumura, T., Sakai, S.i. and Shinohara, M. (2011) 'Significant tsunami observed at ocean-bottom pressure gauges during the 2011 off the Pacific coast of Tohoku Earthquake', *Earth, Planets and Space*, 63(7), pp. 803-808.

Mai, P.M. and Beroza, G.C. (2002) 'A spatial random field model to characterize complexity in earthquake slip', *Journal of Geophysical Research: Solid Earth*, 107(B11).

Manley, J. and Hine, G. (2011) 'Persistent unmanned surface vehicles for subsea support', *Offshore Technology Conference*. Houston, Texas, USA 2-5 May.

Mansinha, L. and Smylie, D.E. (1971) 'The displacement fields of inclined faults', *Bulletin of the Seismological Society of America*, 61(5), pp. 1433-1440.

Matsumoto, H. (2011) *Advances for Tsunami Measurement Technologies and Its Applications*. Japan Agency for Marine-Earth Science and Technology: INTECH Open Access Publisher.

Matsuyama, M. and Tanaka, H. (2001) 'An experimental study of the highest run-up height in the 1993 Hokkaido Nansei-oki earthquake tsunami', *ITS 2001, Proceedings session 7, Number 7-21*.

Melgar, D., Allen, R.M., Riquelme, S., Geng, J., Bravo, F., Baez, J.C., Parra, H., Barrientos, S., Fang, P. and Bock, Y. (2016) 'Local tsunami warnings: Perspectives from recent large events', *Geophysical Research Letters*.

Melgar, D. and Bock, Y. (2013) 'Near-field tsunami models with rapid earthquake source inversions from land- and ocean-based observations: The potential for forecast and warning', *Journal of Geophysical Research-Solid Earth*, 118(11), pp. 5939-5955.

Morales Maqueda, M.A., Penna, N.T., Williams, S.D.P., Foden, P.R., Martin, I. and Pugh, J. (2016) 'Water Surface Height Determination with a GPS Wave Glider: A Demonstration in Loch Ness, Scotland', *Journal of Atmospheric and Oceanic Technology*, 33(6), pp. 1159-1168.

Mori, N., Takahashi, T., Yasuda, T. and Yanagisawa, H. (2011) 'Survey of 2011 Tohoku earthquake tsunami inundation and run-up', *Geophysical Research Letters*, 38.

Neal, J., Schumann, G., Fewtrell, T., Budimir, M., Bates, P. and Mason, D. (2011) 'Evaluating a new LISFLOOD-FP formulation with data from the summer 2007 floods in Tewkesbury, UK', *Journal of Flood Risk Management*, 4(2), pp. 88-95.

Nicolsky, D.J., Suleimani, E.N. and Hansen, R.A. (2011) 'Validation and Verification of a Numerical Model for Tsunami Propagation and Runup', *Pure and Applied Geophysics*, 168(6-7), pp. 1199-1222.

Nikolos, I.K. and Delis, A.I. (2009) 'An unstructured node-centered finite volume scheme for shallow water flows with wet/dry fronts over complex topography', *Computer Methods in Applied Mechanics and Engineering*, 198(47-48), pp. 3723-3750.

Nishiura, D., Furuichi, M. and Sakaguchi, H. (2015) 'Computational performance of a smoothed particle hydrodynamics simulation for shared-memory parallel computing', *Computer Physics Communications*, 194, pp. 18-32.

NVIDIA (2012a) *Adaptive Parallel Computation with CUDA Dynamic Parallelism*. Available at: https://devblogs.nvidia.com/parallelforall/introduction-cuda-dynamic-parallelism/ (Accessed: 26/07/2016).

NVIDIA (2012b) *CUDA C Best Practices Guide*. Available at: http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/ (Accessed: 26/07/2016).

NVIDIA (2012c) *CUDA C Programming Guide*. Available at: http://docs.nvidia.com/cuda/cuda-c-programming-guide/ (Accessed: 26/07/2016).

NVIDIA (2012d) *Dynamic Prallelism in CUDA*. Available at: http://developer.download.nvidia.com/assets/cuda/docs/TechBrief_Dynamic_Parallelism_in_CUDA_v2.pdf (Accessed: 26/07/2016).

NVIDIA (2012e) *Next-Generation CUDA Compute Architecture*. Available at: http://www.nvidia.fr/content/PDF/kepler/NV_DS_Tesla_KCompute_Arch_May_2012_LR.pdf (Accessed: 26/07/2016).

NVIDIA (2012f) *White paper NVIDIA's Next Generation CUDA Compute Architecture*. Available at: http://www.nvidia.co.uk/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf (Accessed: 26/07/2016).

Nwogu, O. (1993) 'Alternative form of Boussinesq equations for nearshore wave propagation', *Journal of waterway, port, coastal, and ocean engineering*, 119(6), pp. 618-638.

Ohta, Y., Kobayashi, T., Tsushima, H., Miura, S., Hino, R., Takasu, T., Fujimoto, H., Iinuma, T., Tachibana, K. and Demachi, T. (2012) 'Quasi real ‐ time fault model estimation for near ‐ field tsunami forecasting based on RTK ‐ GPS analysis: Application to the 2011 Tohoku ‐ Oki earthquake (Mw 9.0)', *Journal of Geophysical Research: Solid Earth*, 117(B2).

Oishi, Y., Imamura, F. and Sugawara, D. (2015) 'Near ‐ field tsunami inundation forecast using the parallel TUNAMI ‐ N2 model: Application to the 2011 Tohoku ‐ Oki earthquake combined with source inversions', *Geophysical Research Letters*, 42(4), pp. 1083-1091.

Okada, Y. (1985) 'Surface deformation due to shear and tensile faults in a half-space', *Bulletin of the Seismological Society of America*, 75(4), pp. 1135-1154.

Ozawa, S., Nishimura, T., Suito, H., Kobayashi, T., Tobita, M. and Imakiire, T. (2011) 'Coseismic and postseismic slip of the 2011 magnitude-9 Tohoku-Oki earthquake', *Nature*, 475(7356), pp. 373-376.

Park, J., Song, T.-R.A., Tromp, J., Okal, E., Stein, S., Roult, G., Clevede, E., Laske, G., Kanamori, H. and Davis, P. (2005) 'Earth's free oscillations excited by the 26 December 2004 Sumatra-Andaman earthquake', *Science*, 308(5725), pp. 1139-1144.

Pau, J.C. and Sanders, B.F. (2006) 'Performance of parallel implementations of an explicit finite-volume shallow-water model', *Journal of Computing in Civil Engineering*, 20(2), pp. 99-110.

Pearson, R.V., Causon, D.M. and Mingham, C.G. (1997) *Simulation of coastal and estuarine hydrodynamics using a high-resolution finite-volume technique on a quadtree Cartesian mesh*. Southampton: Computational Mechanics Publications Ltd.

Percival, D.B., Denbo, D.W., Eblé, M.C., Gica, E., Mofjeld, H.O., Spillane, M.C., Tang, L. and Titov, V.V. (2011) 'Extraction of tsunami source coefficients via inversion of DART buoy data', *Natural Hazards*, 58(1), pp. 567-590.

Pollitz, F.F., Bürgmann, R. and Banerjee, P. (2011) 'Geodetic slip model of the 2011 M9. 0 Tohoku earthquake', *Geophysical Research Letters*, 38(7).

Pophet, N., Kaewbanjak, N., Asavanant, J. and Ioualalen, M. (2011) 'High grid resolution and parallelized tsunami simulation with fully nonlinear Boussinesq equations', *Computers & Fluids*, 40(1), pp. 258-268.

Popinet, S. (2003) 'Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries', *Journal of Computational Physics*, 190(2), pp. 572-600.

Popinet, S. (2011) 'Quadtree-adaptive tsunami modelling', *Ocean Dynamics*, 61(9), pp. 1261-1285.

Popinet, S. (2012) 'Adaptive modelling of long-distance wave propagation and fine-scale flooding during the Tohoku tsunami', *Nat. Hazards Earth Syst. Sci*, 12(4), pp. 1213-1227.

Popinet, S. (2015) 'A quadtree-adaptive multigrid solver for the Serre–Green–Naghdi equations', *Journal of Computational Physics*, 302, pp. 336-358.

Power, W. (2013) 'Review of tsunami hazard in New Zealand (2013 update)', *GNS Science Consultancy Report 2013*, 131, p. 222.

Raghavan, H.K. and Vadhiyar, S.S. (2015) 'Adaptive executions of hyperbolic block-structured AMR applications on GPU systems', *International Journal of High Performance Computing Applications*, 29(2), pp. 135-153.

Roe, P.L. (1986) 'Characteristic-based schemes for the Euler equations', *Annual review of fluid mechanics*, 18(1), pp. 337-365.

Rogers, B. and Dalrymple, R.A. (2008) 'SPH modeling of tsunamis', *Advances in Coastal and Ocean Engineering*, 10, pp. 75-100.

Rogers, B., Fujihara, M. and Borthwick, A.G.L. (2001) 'Adaptive Q - tree Godunov - type scheme for shallow water equations', *International Journal for Numerical Methods in Fluids*, 35(3), pp. 247-280.

Rossinelli, D., Hejazialhosseini, B., Spampinato, D.G. and Koumoutsakos, P. (2011) 'Multicore/multi-GPU accelerated simulations of multiphase compressible flows using wavelet adapted grids', *SIAM Journal on Scientific Computing*, 33(2), pp. 512-540.

Saito, T., Ito, Y., Inazu, D. and Hino, R. (2011) 'Tsunami source of the 2011 Tohoku - Oki earthquake, Japan: Inversion analysis based on dispersive tsunami simulations', *Geophysical Research Letters*, 38(7).

Sanders, J. and Kandrot, E. (2012) *CUDA by example: an introduction to general-purpose GPU programming*. Michigan: Addison-Wesley.

Satake, K., Fujii, Y., Harada, T. and Namegaya, Y. (2013) 'Time and space distribution of coseismic slip of the 2011 Tohoku earthquake as inferred from tsunami waveform data', *Bulletin of the seismological society of America*, 103(2B), pp. 1473-1492.

Sato, M., Ishikawa, T., Ujihara, N., Yoshida, S., Fujita, M., Mochizuki, M. and Asada, A. (2011) 'Displacement above the hypocenter of the 2011 Tohoku-Oki earthquake', *Science*, 332(6036), pp. 1395-1395.

Schive, H.-Y., Tsai, Y.-C. and Chiueh, T. (2010) 'GAMER: a graphic processing unit accelerated adaptive-mesh-refinement code for astrophysics', *The Astrophysical Journal Supplement Series*, 186(2), p. 457.

Schive, H.Y., Zhang, U.H. and Chiueh, T. (2012) 'Directionally unsplit hydrodynamic schemes with hybrid MPI/OpenMP/GPU parallelization in AMR', *International Journal of High Performance Computing Applications*, 26(4), pp. 367-377.

Schreiber, M. (2014) *Cluster-Based Parallelization of Simulations on Dynamically Adaptive Grids and Dynamic Resource Management*. PhD thesis. Universität München.

Shao, G., Li, X., Ji, C. and Maeda, T. (2011) 'Focal mechanism and slip history of the 2011 M w 9.1 off the Pacific coast of Tohoku Earthquake, constrained with teleseismic body and surface waves', *Earth, planets and space*, 63(7), pp. 559-564.

Shi, F., Kirby, J.T., Harris, J.C., Geiman, J.D. and Grilli, S.T. (2012) 'A high-order adaptive time-stepping TVD solver for Boussinesq modeling of breaking waves and coastal inundation', *Ocean Modelling*, 43–44, pp. 36-51.

Shuto, N. (1991) 'Numerical simulation of tsunamis—Its present and near future', *Natural Hazards*, 4(2-3), pp. 171-191.

Shuto, N. and Fujima, K. (2009) 'A short history of tsunami research and countermeasures in Japan', *Proceedings of the Japan Academy. Series B, Physical and Biological Sciences*, 85(8), pp. 267-275.

Shuto, N., Goto, C. and Imamura, F. (1990) 'Numerical simulation as a means of warning for near field tsunamis', *Coastal Engineering in Japan*, 33(2), pp. 173-193.

Simons, M., Minson, S.E., Sladen, A., Ortega, F., Jiang, J., Owen, S.E., Meng, L., Ampuero, J.-P., Wei, S. and Chu, R. (2011) 'The 2011 magnitude 9.0 Tohoku-Oki earthquake: Mosaicking the megathrust from seconds to centuries', *science*, 332(6036), pp. 1421-1425.

Sitanggang, K.I. and Lynett, P. (2005) 'Parallel computation of a highly nonlinear Boussinesq equation model through domain decomposition', *International journal for numerical methods in fluids*, 49(1), pp. 57-74.

Smith, L.S. and Liang, Q. (2013) 'Towards a generalised GPU/CPU shallow-flow modelling tool', *Computers & Fluids*, 88(0), pp. 334-343.

Son, S., Lynett, P.J. and Kim, D.-H. (2011) 'Nested and multi-physics modeling of tsunami evolution from generation to inundation', *Ocean Modelling*, 38(1), pp. 96-113.

Song, S., Zhu, A., Zhang, S., Zheng, L., Yuen, D.A. and Lu, Z. (2012) 'Near Field High Resolution Numerical Simulation of Tohoku 2011 Tsunami on Multi-core System', *Proceedings of the 2012 International Conference on Industrial Control and Electronics Engineering*. IEEE Computer Society, pp. 1881-1884.

Stein, S. and Okal, E.A. (2005) 'Seismology: Speed and size of the Sumatra earthquake', *Nature*, 434(7033), pp. 581-582.

Stelling, G. and Zijlema, M. (2003) 'An accurate and efficient finite ‐ difference algorithm for non ‐ hydrostatic free ‐ surface flow with application to wave propagation', *International Journal for Numerical Methods in Fluids*, 43(1), pp. 1-23.

Sugawara, D., Goto, K. and Jaffe, B.E. (2014) 'Numerical models of tsunami sediment transport—Current understanding and future directions', *Marine Geology*, 352, pp. 295-320.

Süß, M. and Leopold, C. (2008) 'Common Mistakes in OpenMP and How to Avoid Them', in Mueller, M., Chapman, B., de Supinski, B., Malony, A. and Voss, M. (eds.) *OpenMP Shared Memory Parallel Programming*. Springer Berlin Heidelberg, pp. 312-323.

Sweby, P.K. (1984) 'High resolution schemes using flux limiters for hyperbolic conservation laws', *SIAM journal on numerical analysis*, 21(5), pp. 995-1011.

Synolakis, C.E., Bernard, E.N., Titov, V.V., Kânoğlu, U. and González, F.I. (2008) 'Validation and Verification of Tsunami Numerical Models', *Pure and Applied Geophysics*, 165(11), pp. 2197-2228.

Tang, L., Titov, V.V., Bernard, E.N., Wei, Y., Chamberlin, C.D., Newman, J.C., Mofjeld, H.O., Arcas, D., Eble, M.C. and Moore, C. (2012) 'Direct energy estimation of the 2011 Japan tsunami using deep ‐ ocean pressure measurements', *Journal of Geophysical Research: Oceans*, 117(C8).

Tang, L., Titov, V.V. and Chamberlin, C.D. (2009) 'Development, testing, and applications of site ‐ specific tsunami inundation models for real ‐ time forecasting', *Journal of Geophysical Research: Oceans*, 114(C12).

Tanioka, Y. and Satake, K. (1996) 'Tsunami generation by horizontal displacement of ocean bottom', *Geophysical Research Letters*, 23(8), pp. 861-864.

Tappin, D.R., Grilli, S.T., Harris, J.C., Geller, R.J., Masterlark, T., Kirby, J.T., Shi, F., Ma, G., Thingbaijam, K.K.S. and Mai, P.M. (2014) 'Did a submarine landslide contribute to the 2011 Tohoku tsunami?', *Marine Geology*, 357, pp. 344-361.

Tatehata, H. (1997) 'The new tsunami warning system of the Japan Meteorological Agency', in *Perspectives on Tsunami Hazard Reduction*. Springer, pp. 175-188.

Titov, V.V. (2009) 'Tsunami forecasting', *The sea*, 15, pp. 371-400.

Titov, V.V., Gonzalez, F.I., Bernard, E.N., Eble, M.C., Mofjeld, H.O., Newman, J.C. and Venturato, A.J. (2005) 'Real-time tsunami forecasting: Challenges and solutions', *Natural Hazards*, 35(1), pp. 41-58.

Titov, V.V. and Synolakis, C.E. (1995) 'Modeling of breaking and nonbreaking long-wave evolution and runup using VTCS-2', *Journal of Waterway, Port, Coastal and Ocean Engineering - ASCE*, 121(6), pp. 308-316.

Titov, V.V. and Synolakis, C.E. (1998) 'Numerical modeling of tidal wave runup', *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 124(4), pp. 157-171.

Toro, E.F. (2001) *Shock-Capturing Methods for Free-Surface Shallow Flows*. Chichester: Wiley.

Toro, E.F. (2009) *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer Berlin Heidelberg.

Toro, E.F., Spruce, M. and Speares, W. (1994) 'Restoration of the contact surface in the HLL-Riemann solver', *Shock Waves*, 4(1), pp. 25-34.

Tsushima, H., Hino, R., Fujimoto, H., Tanioka, Y. and Imamura, F. (2009) 'Near - field tsunami forecasting from cabled ocean bottom pressure data', *Journal of Geophysical Research: Solid Earth*, 114(B6).

Vacondio, R., Dal Palù, A. and Mignosa, P. (2014) 'GPU-enhanced finite volume shallow water solver for fast flood simulations', *Environmental Modelling and Software*, 57, pp. 60-75.

Vacondio, R., Rogers, B.D. and Stansby, P.K. (2012) 'Accurate particle splitting for smoothed particle hydrodynamics in shallow water with shock capturing', *International Journal for Numerical Methods in Fluids*, 69(8), pp. 1377-1410.

Valdez-Balderas, D., Domínguez, J.M., Rogers, B.D. and Crespo, A.J.C. (2013) 'Towards accelerating smoothed particle hydrodynamics simulations for free-surface flows on multi-GPU clusters', *Journal of Parallel and Distributed Computing*, 73(11), pp. 1483-1493.

van Leer, B. (1985) 'On the relation between the upwind-differencing schemes of Godunov, Enguist-Osher and Roe', *SIAM Journal on Scientific and Statistical Computing*, 5(1), pp. 1-20.

Vazhenin, A., Lavrentiev, M., Romanenko, A. and Marchuk, A. (2013) 'Acceleration of tsunami wave propagation modeling based on re-engineering of computational components', *International Journal of Computer Science and Network Security*, 13(3), pp. 32-40.

Vijayasherly, V., Shalini, L., Sharma, C. and Pandya, M. (2016) 'Implementation of Load Balancing Algorithm for Data Intensive Applications using MPI', *International Journal of Emerging Engineering Research and Technology*, 4(6), pp. 1-10.

Wang, P., Abel, T. and Kaehler, R. (2010) 'Adaptive mesh fluid simulations on GPU', *New Astronomy*, 15(7), pp. 581-589.

Wang, X. (2008) *Numerical modelling of surface and internal waves over shallow and intermediate water*. PhD thesis. Cornell University.

Wang, X. (2009) 'User manual for COMCOT version 1.7 (first draft)'. Cornell University.

Wang, X. and Liu, P.L.F. (2006) 'An analysis of 2004 Sumatra earthquake fault plane mechanisms and Indian Ocean tsunami', *Journal of Hydraulic Research*, 44(2), pp. 147-154.

Wang, Y., Liang, Q., Kesserwani, G. and Hall, J.W. (2011) 'A 2D shallow flow model for practical dam-break simulations', *Journal of Hydraulic Research*, 49(3), pp. 307-316.

Watanabe, Y., Mitobe, Y., Saruwatari, A., Yamada, T. and Niida, Y. (2012) 'Evolution of the 2011 Tohoku earthquake tsunami on the Pacific coast of Hokkaido', *Coastal Engineering Journal*, 54(01), p. 1250002.

Wei, G., Kirby, J.T., Grilli, S.T. and Subramanya, R. (1995) 'A fully nonlinear Boussinesq model for surface waves. Part 1. Highly nonlinear unsteady waves', *Journal of Fluid Mechanics*, 294, pp. 71-92.

Wei, Y., Chamberlin, C., Titov, V.V., Tang, L. and Bernard, E.N. (2013) 'Modeling of the 2011 Japan Tsunami: Lessons for Near-Field Forecast', *Pure and Applied Geophysics*, 170(6-8), pp. 1309-1331.

Wei, Y., Cheung, K.F., Curtis, G.D. and McCreery, C.S. (2003) 'Inverse algorithm for tsunami forecasts', *Journal of waterway, port, coastal, and ocean engineering*, 129(2), pp. 60-69.

Wei, Y., Newman, A.V., Hayes, G.P., Titov, V.V. and Tang, L. (2014) 'Tsunami forecast by joint inversion of real-time tsunami waveforms and seismic or GPS data: application to the Tohoku 2011 tsunami', *Pure and Applied Geophysics*, 171(12), pp. 3281-3305.

Wilson, R.I., Admire, A.R., Borrero, J.C., Dengler, L.A., Legg, M.R., Lynett, P., McCrink, T.P., Miller, K.M., Ritchie, A. and Sterling, K. (2013) 'Observations and impacts from the 2010 Chilean and 2011 Japanese tsunamis in California (USA)', *Pure and Applied Geophysics*, 170(6-8), pp. 1127-1147.

Xia, X. and Liang, Q. (2016) 'A GPU-accelerated smoothed particle hydrodynamics (SPH) model for the shallow water equations', *Environmental Modelling & Software*, 75, pp. 28-43.

Yamazaki, Y., Cheung, K.F. and Kowalik, Z. (2011a) 'Depth-integrated, non-hydrostatic model with grid nesting for tsunami generation, propagation, and run-up', *International Journal for Numerical Methods in Fluids*, 67(12), pp. 2081-2107.

Yamazaki, Y., Cheung, K.F. and Lay, T. (2013) 'Modeling of the 2011 Tohoku Near‐Field Tsunami from Finite‐Fault Inversion of Seismic Waves', *Bulletin of the Seismological Society of America*, 103(2B), pp. 1444-1455.

Yamazaki, Y., Cheung, K.F., Pawlak, G. and Lay, T. (2012) 'Surges along the Honolulu coast from the 2011 Tohoku tsunami', *Geophysical Research Letters*, 39(9).

Yamazaki, Y., Kowalik, Z. and Cheung, K.F. (2009) 'Depth‐integrated, non‐hydrostatic model for wave breaking and run‐up', *International Journal for Numerical Methods in Fluids*, 61(5), pp. 473-497.

Yamazaki, Y., Lay, T., Cheung, K.F., Yue, H. and Kanamori, H. (2011b) 'Modeling near‐field tsunami observations to improve finite‐fault slip models for the 11 March 2011 Tohoku earthquake', *Geophysical Research Letters*, 38(7).

Yiu, K.F.C., Greaves, D.M., Cruz, S., Saalehi, A. and Borthwick, A.G.L. (1996) 'Quadtree grid generation: Information handling, boundary fitting and CFD applications', *Computers and Fluids*, 25(8), pp. 759-769.

Yoon, S.B. (2002) 'Propagation of distant tsunamis over slowly varying topography', *Journal of Geophysical Research: Oceans*, 107(C10).

Yoon, S.B., Lim, C.H. and Choi, J. (2007) 'Dispersion-correction finite difference model for simulation of transoceanic tsunamis', *Terrestrial Atmospheric and Oceanic Sciences*, 18(1), p. 31.

Zhang, S., Yuen, D.A., Zhu, A., Song, S. and George, D.L. (2011) 'Parallelization of GeoClaw Code for Modeling Geophysical Flows with Adaptive Mesh Refinement on Many-core Systems', *Computational Science and Engineering (CSE), 2011 IEEE 14th International Conference on*. 24-26 Aug. 2011. pp. 573-579.

Zhang, Y.J. and Baptista, A.M. (2008) 'An efficient and robust tsunami model on unstructured grids. Part I: Inundation benchmarks', *Pure and Applied Geophysics*, 165(11-12), pp. 2229-2248.

Zhou, H., Wei, Y. and Titov, V.V. (2012) 'Dispersive modeling of the 2009 Samoa tsunami', *Geophysical Research Letters*, 39(16).

Zoppou, C. and Roberts, S. (2003) 'Explicit schemes for dam-break simulations', *Journal of Hydraulic Engineering*, 129(1), pp. 11-34.